Context Awareness and Trust 2008

2nd International Workshop on
Combining Context with Trust, Security and Privacy

(Co-located with IFIPTM 2008, 16th June 2008, Trondheim - Norway)

Proceedings

Editors

Bob Hulsebosch (Telematica Instituut)
Gabriele Lenzini (Telematica Instituut)
Jean-Marc Seigneur (University of Geneva and Venyo)
Santtu Toivonen (Idean)

Preface

This volume contains the proceedings of the 2nd International Workshop on Context Awareness and Trust (CAT08) held on June 16th 2008 in Trondheim -Norway, and *co-located* the joint iTrust and PST conferences on Trust Management and Security (IFIPTM 2008).

As last year, the CAT08 workshop continues to stimulate an active exchange of new ideas on the bidirectional relationship between the area of *context awareness* and the area of *trust, security, and privacy*; it aims to bring experts together, to collect the state of the art, to identify open and emerging problems, and to propose future research directions.

The evolution of today's infrastructures towards more pervasive and ubiquitous context aware infrastructures raises new *challenges* and *new opportunities* regarding trust, security, and privacy. The focus on combining trust, security, and privacy with context awareness will bring new insights as well as provide an interesting playground for the IFIPTM community. In particular:

- The opportunity to *use context as an approach* to enhance privacy, trust or security seems an interesting, innovative, and value-adding extension to IFIPTM. For example, the availability of context information can help the establishment of trust relationship (users in the same room for the same meeting are more willing to trust each other more than users without visible contact). Moreover, contextual information can be used to improve *dynamic, adaptive, and autonomic aspects* of security, access control, and privacy control/enforcement.

- The opportunity to apply the results achieved in security, privacy, and trust to *strengthen context aware infrastructures and to facilitate the exchange of context information* is a challenge for IFIPTM community. For example, because context information often has a personal character, privacy and other rights of individuals must be carefully protected. Moreover, trust is an essential prerequisite for secure exchange and usage of context information at different quality levels.

Following the trend initiated last year, the CAT08 workshop has attracted the attention of researchers with a computer science and information & communication technology background, whose experiences root both in the academia and the industry.

In the current edition, four papers have been accepted for oral presentations out of a pool of six papers. Papers were reviewed by at least three reviewers of the program committee, and all selected papers are of high quality.

The articles in of this volume are representative of the current research activities on the topics of the workshop. Their contributions focus on the prediction of trust relationship using contextual information, on the analysis of requirements for identity management architectures, on approaches for privacy preserving in location-aware applications for mobile social networks and, finally, on the analysis of user behaviour with respect to privacy aspects in location-aware systems for social Interaction.

An on-line version of the present proceedings will be published by CEUR.org; it will also be available at the workshop website: http://cat08.telin.nl.

We would like to take this opportunity to thank people who contributed to the CAT08 workshop. We wish to thank the PC members, the reviewers, and in particular the authors for their valuable contributions. We wish them a successful continuation of their work in this area. Finally, we thank the organization of the IFIPTM 2008 conference with which this workshop was co-located.

May 2008

Bob Hulsebosch,
Gabriele Lenzini,
Jean-Marc Seigneur,
Santtu Toivonen.

Organization

- Bob Hulsebosch (Telematica Instituut, The Netherlands)
- Gabriele Lenzini (Telematica Instituut, The Netherlands)
- Jean-Marc Seigneur (University of Geneva and Venyo, Switzerland)
- Santtu Toivonen (Idean, Finland)

Program Committee

- Susana Alcalde (University of Navarra, Spain)
- Claudio Bettini (University of Milano, Italy)
- Harry Chen (Image Matters LLC, USA)
- Tyrone W. Grandison (IBM Almaden Research, USA)
- Victor S. Grishchenko (Ural State University, Russia)
- Heikki Helin (TeliaSonera, Finland)
- Mario Hoffmann (Fraunhofer-Institute SIT, Germany)
- Bob Hulsebosch (Telematica Instituut, The Netherlands)
- Aman Kayssi (University of Beirut, Lebanon)
- Anders Kofod-Petersen (NTNU, Norway)
- Gabriele Lenzini (Telematica Instituut, The Netherlands)
- Giannis F. Marias (Athens University of Economics and Business, Greece)
- Daniel Olmedilla (L3S Research Center and University of Hannover, Germany)
- Daniele Quercia (University College London, United Kingdom)
- Bart van Rijnsoever (Philips Research, The Netherlands)
- Philip Robinson (SAP Research, United Kingdom)
- Jean-Marc Seigneur (University of Geneva and Venyo, Switzerland)
- Santtu Toivonen (Idean, Finland)
- Maarten Wegdam (University of Twente, The Netherlands)
- Zheng Yan (Nokia Research, Finland)

Supporting Organizations

Freeband Communication, Telematica Instituut, University of Geneva, Venyo, Idean.

Table of Contents

Implementing Privacy as Symmetry in Location-aware Systems

Anders Kofod-Petersen, Espen Klæboe, Jørgen Jervidalo, Kjetil Aaltvedt, Magnus Romnes, and Trond Martin Nyhus

Department of Computer and Information Science,
Norwegian University of Science and Technology,
7491 Trondheim, Norway
anderpe@idi.ntnu.no,{espenkl|jervidal|kjetilue|romnes|trondmn}@stud.ntnu.no

Abstract. Social network services on the internet are moving from a traditional web-based service into ubiquitous computing environments. With this migration these services will also benefit from the context-awares that ubiquitous computing offers. Applications that sense the environment and act proactively, requires an immaculate attention to users' privacy. The work presented here approaches privacy by employing the *principle of minimum asymmetry* to a mobile social network service. We demonstrate how this principle can be implemented on a simple location-aware application running on standard mobile telephones in a traditional GSM network.

1 Introduction

The number and popularity of digital social networks have been steadily increasing over the last few years. Privacy and trust are important issues in these social aware applications. The importance will increase tremendously when social aware applications are moved into mobile or pervasive applications. When applications assume responsibility from the user and act proactively, as pervasive applications do by definition, the control over what information is shared and to whom becomes paramount.

One important approach to maintain privacy and trust in pervasive applications is the principle of minimal asymmetry, which in short states that the ability to obtain information should be coupled with the sharing of information. The work presented here demonstrates how how this principle can be applied in a mobile social-aware application. The application implements location-awareness on standard mobile phones in running in a GSM network.

The rest of the paper is organised as follows: First, an overview of related work concerning privacy in ubiquitous computing is presented. This is followed by a description of the systems design and implementation. The paper ends with a summary and pointers to future work.

2 Related Work

Social systems that link people to people, and people to geographical places are referred to as P3 systems [1]. P3 systems can be divided into two difference categories: *people-centered* and *place-centered*. An example of a people-centered system might be one where a user has a contact list, where the contacts show up in different colours (green, yellow, red) depending on their proximity. Jones and Grandhi presented a survey executed at various places in Manhattan with more that 500 participants. Among other things, they discovered that 84% of the participants were willing to share their location data (anonymously) to get information about crowding and occupancy in public places. They concluded that a large population considered P3 systems to be sufficiently beneficial to disclose their position. The fact that this percentage of the population were willing to give away their position in exchange for a service that they considered beneficial is an important insight when modelling context-aware systems, and perhaps even more important when systems are to reason about what (contextual) information they can share and to whom.

With regard to privacy in context-aware systems, Langheinrich [2] describes why privacy is of particular importance in ubiquitous computing with four properties: *ubiquity*, computers are everywhere; *invisibility*, computers disappear from the scene; *sensing*, sensors are becoming more precise; and *memory amplification*, storing of large amount of (sensed) data.

Many of the privacy issues in context-aware systems are related to the issue of *mutual awareness*. One part of this problem is about *disembodiment* and *dissociation*. When we encounter people in the real world we can receive information in many ways, such as position, voice level, face expression and direction of gaze. In ubiquitous environments these communication channels are likely to be less effective. In real life people live by the intuitive principle that if you cannot see me, I cannot see you. Due to the potential large number of sensors in an ubiquitous environment, this is not always true. Users may not always know exactly what information they are conveying, in what form, if it is permanent, and to whom they are sending [3].

Jiang et al. [4] discusses the *principle of minimum asymmetry* when dealing with privacy issues in ubiquitous computing. This principle goes a long way towards handling the apparent asymmetric relationship between sender and receiver of information, as described by Bellotti et al. [3]. Jiang et at. argue that a privacy-aware system should minimise the asymmetry of information between data owners and data users. For an example, if a user does not wish to share his location he cannot expect others to share their location with him (regardless of their wish). The main principle is that [4, p. 7] (original emphasis):

> A privacy aware system should' minimize the asymmetry of information between **data owners** and **data collectors and data users**, by:
> – **Decreasing** the flow of information from data owners to data collectors and users

– **Increasing** the flow of information from data collectors and users back to data owners

When decreasing the flow of information from the data owner the user maintains a higher degree of control over the system. Increasing information flow from the data collector provides better feedback to the user. Examples of mechanisms that adhere to this principle are: *anonymising* or *pseudonymising*, which approaches privacy by allowing users to act in total anonymity or through a consistent avatar that cannot be coupled with a real person. Further, *plausible deniability* allows the user to plausible deny that he did not wish to interact with a particular person, at a given time [5]. Finally, *reciprocity* is the essential property for building trust and deep relationships. Sharing too little or too much information might have a negative effect on relations to one's peers. Balancing the amount on information flowing between peers are very important to maintain a balance in any relationship. Social systems often approaches this by for an example only allowing you to see the status of the people whom you allow to see your status. To receive better feedback from the system it might log all access to one's position data, notify the user when somebody requests a position, and give clear feedback on what information is stored [4].

Lederer et al. [6] argues that feedback and control is "the designer's opportunity to empower those processes (understanding and action), and they are the user's opportunity to practice them." The authors exemplify pitfalls in design of systems maintaining privacy using their personal experience. The pitfalls can be divided into two main groups: feedback and control. The feedback pitfalls are: *obscuring potential information flow*, where systems do not explicitly describe the possible disclosures it can make; and *obscuring actual information flow*, where a system might not explicitly make clear what information is actually disclosed. The control pitfalls are: *emphasising configuration over action*, where configuration overshadows the privacy management actually needed to adapt to a user's ordinary use of the system; *lacking coarse-grained control*, where a system offers too many choices and not just simple on/off choices; and *Inhibiting existing practice*, where a system forces some required practice onto a user, and does not adapt to the user's practice.

As aforementioned, digital social networks have recently emerged, and along with them several communication protocols and representational models. Among these are: XHTML Friends Network (XFN), Friend of a Friend (FOAF) and Extensible Messaging and Presence Protocol (XMPP) commonly known as Jabber.

XFN[1] is a decentralised markup language that captures social connectivity. This language can represent different forms of social connectivity, such as the level of friendship, professional relations, geographical proximity, family ties and romantic aspects. XFN maintains some privacy related limitations, such as the fact that many personal attributes cannot be assigned by others. Some examples of these include: gender, race and age. This limitation is founded in the fact that a user describes a *relation* to another person from that user's perspective, and

[1] http://gmpg.org/xfn/

not the friend. Since a “friendship” in XFN is directional relationship managed solely by the user who initiates a relation, the principle of minimal asymmetry seems hard to maintain.

FOAF[2] is like XFN a modelling language used to represent social relations. FOAF have chosen RDF, a more expressive language that XHTML chosen by XFN. FOAF is currently an immature language, which does not explicitly cover issues such as control of data within a community of trust, guaranty of facts and general privacy of data.

XMPP[3], or Jabber as it is commonly know, is a message protocol for instant messaging services [7,8]. The Jabber specification defines a XML streaming protocol that covers not only instant messaging but also issues such as presence. As the XML protocol is extensible, the core functionality such as authentication, privacy mechanisms and chatting are easily extensible.

3 Design and Implementation

Find Per Anton (hereby referred to as ”the application”) is a location-aware instant messaging application developed for mobile phones. It utilises the underlying TCP/IP-network capabilities (such as GRPS, UMTS and Wi-Fi) of the mobile phone and will supplement a similar application based on Wi-Fi technology [9]. The localisation service is obtained from the third party provider Geomatikk[4] and all transmissions between the server and the client uses the XMMP protocol.

The idea is to have the opportunity to sort your friends in groups, locate one friend or a whole group on the map and also be able to send messages to contacts or whole groups. The application provides feedback about the locations of the peers of a user by indicating their proximity using colours ranging from red to green and by showing their position on a map. An example of how the map service might look can be seen in Fig. 1.

Table 1. Presence states allowed by the XMPP standard.

Name:	**Description:**
offline	The entity or resource is not connected to the server.
chat	The entity or resource is actively interested in chatting.
away	The entity or resource is temporarily away.
dnd	The entity or resource is busy (dnd = ”Do Not Disturb”).
xa	The entity or resource is away for an extended period (xa = ”eXtended Away”).

[2] http://www.foaf-project.org/
[3] http://www.xmpp.org/
[4] http://www.geomatikk.no/

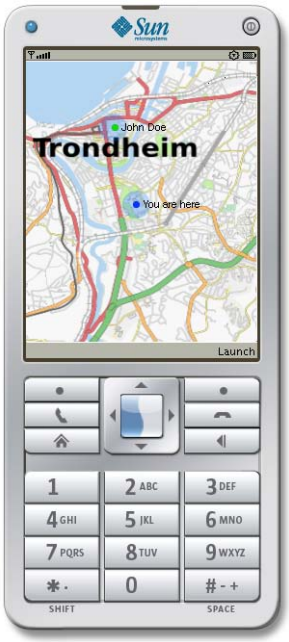

Fig. 1. An example of the map service running on a mobile device

As mentioned above the application design builds on the XMPP standards [7,8]. Unfortunately, XMPP does not enforce the principle of minimum assymmetry, which is a requirement for the application. For this reason, a deviation was made between the original specification and the application design by only supporting a subset of the XMPP subscription states. The subscription states supported by the original XMPP specification can be seen in Table 2. Every subscription state except *None* and *Both* was omitted from the application design because they contradict the principle of minimum asymmetry by allowing a user to receive information without releasing information himself. In the XMPP standard, adding a contact is refered to as *subscribing to a contact* and a contact list is refered to as a *roster list*, so we will use these terms from now on.

In order to establish a subscription, both users must agree to share *presence information* and *location information* with the other part. This requirement allows the application to enforce the principle of minimum asymmetry. The presence information contains the current status of the user, which may be *away*, *chat*, *dnd*, *xa* or *offline*. A detailed description of the presence status can be found in Table 1. The location information contains the longitude and latitude of a user, and will appear graphically as a position on a map. This way a user can physically locate the position of any of his contacts, in return for being discoverable himself. A user can temporarily make himself invisible to others, but he is then no longer entitled to receive any location information, until he

Table 2. Subscription states supported by the XMPP standard, described from the user's perspective.

Name:	**Description:**
"None"	Contact and user are not subscribed to each other, and neither has requested a subscription from the other.
"None + Pending Out"	Contact and user are not subscribed to each other, and user has sent contact a subscription request but contact has not replied yet.
"None + Pending In"	Contact and user are not subscribed to each other, and contact has sent user a subscription request but user has not replied yet.
"None + Pending Out/In"	Contact and user are not subscribed to each other, contact has sent user a subscription request but user has not replied yet, and user has sent contact a subscription request but contact has not replied yet.
"To"	User is subscribed to contact (one-way).
"To + Pending In"	User is subscribed to contact, and contact has sent user a subscription request but user has not replied yet.
"From"	Contact is subscribed to user (one-way).
"From + Pending Out"	Contact is subscribed to user, and user has sent contact a subscription request but contact has not replied yet.
"Both"	User and contact are subscribed to each other (two-way).

makes himself discoverable again, as discussed by Jiang et. al [4]. When he makes himself invisible, he will appear as offline to his contacts, just as if he had turned off his mobile phone.

When a user cancels a subscription, the contact is not notified. The contact is removed from the user's roster list, but the user is only shown as offline on the contact's roster list, thus preserving minimum asymmetry and plausible deniability.

In Listing 1.1, we show a simplified pseudo code example of how a user adds a subscriber (contact) to his roster list in the application. The client sends a request for a new subscription to the server, using the method `newSubscription()`. The server checks the current subscription state for the subscriber. In an initial state, there will not exist any relations between the user and the contact, so the server will store the new subscription state (*None*) and send a subscription request to the contact. If the contact accepts the subscription request, his client will send a subscription request to the server, again using the same `newSubscription()`-method. This time, when the server receives the request, there already exists a relation between the user and the contact, so the server sets the subscription states to *Both* for the user and the contact, adds them to the respective roster lists and pushes the new roster lists to both clients. The symmetrical subscription state *Both* is the *only* state where positioning, presence notifications or messaging is allowed.

Listing 1.1. Implementation Pseudo Code (Add a Friend)

```
// [CLIENT] User 1 requests to add User 2 as a friend:
function addFriend(newcontact)
{
  // Send a request for a new subscription to the server:
  server.newSubscription(currentuser, newcontact)
}

// [SERVER] Receiving notification about new subscription
function newSubscription(user, contact)
{
  if contact.hasAccepted(user)
    // Sets the subscription state for both parties and
    // add 'user' to 'contact's roster list:
    setSubscriptionState(contact, user, BOTH)
    setSubscriptionState(user, contact, BOTH)
    addToRoster(contact, user)

  else if contact.hasDenied(user)
    // Don't make the user ask the contact for permission
    // any more:
    setAskForAcceptance(user, contact, FALSE)

  else
    // As long as the contact is undeceided or has refused,
    // we use NONE. However, user has contact on his roster
    // list
    setSubscriptionState(user, contact, NONE)
    addToRoster(user, contact)
    // Ask User 2 (contact) to accept symmetrical subscription
    // to User 1
    askForAcceptance(contact)
}

// [CLIENT] User 2 (currentuser) receives subscription
// request from User 1 (contact)
function onSubscriptionRequestEvent(contact)
{
  // Show GUI asking for acceptance
  if accepted
    // User 2 accepted a symmetrical subscription to User 1
    server.newSubscription(currentuser, contact)
}

// [CLIENT] User 1 receives updated
function onRosterUpdateEvent()
{
  redrawRoster()
}
```

When a user wants to remove a subscriber from his roster list, we must not only enforce the principle of minimum asymmetry, but also retain privacy for the user. In other words, the removal has to be done in a discrete way. Pseudo code showing how this is handled in the system, can be seen in Listing 1.2. The user's client calls the `deleteSubscription()`-method on the server, which removes the specified subscriber from the user's roster list. Note that the user **is not** removed from the subscriber's roster list. Instead the removed contact's subscription state to the user is set to *None* on the server. This will result in the user to always appear offline to the contact.

By applying these methods to the application, we achieve minimum asymmetry both with respect to presence information and location information.

Listing 1.2. Implementation Pseudo Code (Remove a Friend)

```
// [CLIENT]: User 1 wants to delete User 2 (contact)
function deleteSubscription(contact)
{
  server.deleteSubscription(currentuser, contact)
}

// [SERVER]: User 1 wants to delete User 2 (contact)
function deleteSubscription(user, contact)
{
  // Remove contact from user's roster:
  removeFromRoster(user, contact)

  // Set user's subscription state to NONE in contact's
  // roster. Contact will see user as offline.
  setSubscriptionState(contact, user, NONE)

  // Send a new roster to user without contact in it.
  sendUpdatedRoster(user)
}
```

4 Summary and Further work

The work presented here has demonstrated how the *principle of minimum asymmetry* can be applied as a means of maintaining privacy in a mobile social application. The service uses the XMPP protocol to exchange messages between different users. However, the XMPP standard does not conform to the principle of minimum asymmetry. By using a subset of the standard, we show how we can enforce mutual subscriptions and thus minimum asymmetry.

A similar system has been developed for the Wi-Fi environment in Wireless Trondheim [9]. This system has primarily been developed to investigate any behavioural changes in the users when they *know the location of their peers* and

they know that their peers know their location[5]. We expect to conduct similar experiments on a larger scale, using the implemented system described in this paper. The experiment will be conducted using the GSM mobile network in Norway.

Acknowledgements

Parts of this work has been supported by Accenture Innovation Lab Norway. We would like to extend our thanks to Per Anton Gransæther for his assistance.

References

1. Jones, Q., Grandhi, S.A.: P3 systems: Putting the place back into social networks. IEEE Internet Computing **9** (2005) 38–46
2. Langheinrich, M.: Privacy by design – principles of privacy-aware ubiquitous systems. In Abowd, G.D., Brumitt, B., Shafer, S.A., eds.: Proceedings of the Third International Conference on Ubiquitous Computing (UbiComp 2001). Number 2201 in Lecture Notes in Computer Science, Springer Verlag (2001) 273–291
3. Bellotti, V., Sellen, A.: Design for privacy in ubiquitous environments. In Michelis, G.D., Simone, C., Schmidt, K., eds.: Proceeding of the Third European Conference on Computer-Supported Cooperative Work (ECSCW '93), Kluwer Academic Publishers (1993) 77–92
4. Jiang, X., Hong, J.I., Landay, J.A.: Approximate information flows: Socially-based modeling of privacy in ubiquitous computing. In Borriello, G., Holmquist, L.E., eds.: Proceedings of the 4th International Conference on Ubiquitous Computing (UbiComp 2002). Volume 2498 of Lecture Notes in Computer Science., Springer Verlag (2002) 176–193
5. Raento, M., Oulasvirta, A.: Privacy management for social awareness applications. In Floréen, P., Lindén, G., Niklander, T., Raatikaine, K., eds.: Workshop on Context Awareness for Proactive Systems (CAPS 2005), HIIY Publications (2005) 105–114
6. Lederer, S., Hong, I., Dey, K., Landay, A.: Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing **8** (2004) 440–454
7. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920 (2004)
8. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921 (2004)
9. Andresen, S., Krogstie, J., Jelle, T.: Lab and research activities in wireless trondheim. In: Proceedings of IEEE International Symposium on Wireless Communication Systems, IEEE Computer Society (2007) 385–389

[5] This research is currently being prepared for publication

Towards Context-Enriched Trust Prediction: A Proposal

Marcin Sydow

Polish-Japanese Institute of Information Technology,
Web Mining Lab,
Koszykowa 86, 02-008, Warszawa, Poland**
msyd@pjwstk.edu.pl

Abstract. This short paper describes an early stage of research aiming at improving trust prediction in social networks.
It builds on recent findings on observed correlation between trust and user similarity. A machine-learning approach to the trust prediction problem with use of contextual information is proposed and experimental work envisaged.
An overview of the existing results is presented and it indicates that the proposed approach constitutes a novel combination of ideas and, as such, has a potential to contribute to the previous research in the area.

Key words: trust metrics, context, experimentation, social networks

1 Introduction

There is an explosion of interest in on-line communities where users' personal preferences can be explicitly expressed. Trust management plays an important role in such systems since on-line users base their decisions on information expressed by other users. However, the amount of information contained in social networks to be analysed by the users grows exponentially, so that there is a strong need for developing and improving automatic techniques that support users in making their on-line activities.

An example of such automatic support is a recommender system like `epinions`[1] or `FilmTrust`[2]. The system, given the preferences of the users *and* trust values expressed among them, is capable of automatically recommending some items to particular users. In this way, the information about trust among the users significantly improves the decision-support process.

The mechanism mentioned above can work quite well given that the trust values are expressed by the users. However, since on-line communities grow constantly, significant part of the users are newcomers to the system so that they did not specify their trust to other users, yet.

** The work is supported by the Polish Ministry of Science grant N N516 4307 33.

[1] `http://www.epninions.com`

[2] `http://trust.mindswap.org/FilmTrust/`

Due to this problem, we study the issue which is somehow reverse to the recommendation problem mentioned earlier. Namely, we aim at studying the problem of how to better predict an unknown trust between a pair of users given some additional information available in the system which we call the *contextual* information.

The term *contextual information* in social networks can be interpreted in many ways. In the most general meaning, the context of trust in a social network can be understood as all the information available in the social network except the trust information itself.

In our research, however, we plan to initially narrow the definition of the contextual information to the recommendations (ratings) given by the users. Thus, our notion of the context-aware trust might differ from those considered elsewhere, e.g. in [12] or [10]. This limited understanding of the trust context definition is partially due to the fact that quite often only the recommendation data, except the trust data itself, is available in publicly accessible real datasets concerning trust-related research. In particular, we plan to build *user profiles* based on their recommendations, and subsequently treat such profiles as a special kind of contextual information in the task of trust prediction.

An example of a dataset, with both: trust and recommendation data available, is the epinion dataset [1]. We plan to start the experimentation with this particular dataset.

In future, if the results obtained for such a limited definition of the context of trust are encouraging and we have access to datasets with richer information, we plan to treat the context of trust in a broader meaning.

2 Motivation and Related Work

A recent study by Ziegler et al. [15] experimentally proves, that there exists a significant correlation between the trust expressed by the users and their similarity based on the recommendations they made in the system.

Our proposal of research builds on that finding and considers going a step further. Namely, using the recommendations made by the users as the contextual information we aim at *predicting* potentially unknown trust between users. The main difference is that in the cited paper merely the *correlation* between the trust and the recommendation-based context is measured, while in contrast, we plan to measure how much the context helps in a full machine-learning approach to predicting the unknown trust value.

More precisely, our research aims at experimentally comparing the predictive value of the three groups of attributes in the machine-learning task of the trust prediction in social networks. The three groups of attributes are:

1. the attributes based solely on the pure topology of the trust network
2. the attributes based on the contextual information. As was explained, we plan to begin with taking the recommendation-based information to build such attributes
3. the combined trust-based and contextual-based attributes

One of the main goals of the proposed research is to experimentally check our hypothesis which is as follows: *a trust-predictor build on the third, contextually-enriched group of attributes, significantly outperforms the predictors build on pure-trust and pure-contextual attributes, separately, which seems to be suggested by the results reported in [15]*

A good general introduction to the trust management issues is given in [11]. Trust and distrust propagation models are proposed in [4] and further enhanced in [13] in the special context of Web spam combating.

The problem of pure link-based classification and appropriate link-based features are discussed by Liben-Nowell et al. [7] which proposes some measures for analysing proximity of nodes in a social *undirected* network representing co-authorship. Karamon et al. continues this research [5], however in this work the problem of *node classification* rather than *link classification* is discussed.

Correlation between trust and user profile similarity is discussed in [15] and further studied – by means of survey-based experiments – in [2]. An algorithm based on Bayesian network for trust inference is discussed in [6].

In the time of writing, the author discovered an unpublished draft paper concerning the problem of trust-based rating prediction and, inversely, similarity-based trust prediction by Matsuo et al. [9]. However, the current version of the draft does not seem to be completed (at the time of writing), for example, some important formulae seem to need to be syntactically clarified. However, since there seems to be a remarkable overlap with what is proposed in our research it seems necessary to address the issue how they relate to each other. In particular, in its current form, the mentioned draft:

- proposes general topology-based characteristics for trust computation, which, for example, never propagate further than through two links. In contrast, this paper proposes features that are especially designed to trust (and distrust) propagation measurement and are not constrained in such a way
- unfortunately does not address the problem of the extreme rating-data sparsity. In practice, in most of the available datasets the ratings data is too sparse to be directly processed. Our approach is aware of this issue and proposes how to try to overcome this problem
- does not seem to take into account the distrust concept

It would be very interesting, though, to compare the final results of the mentioned paper, and the research envisaged here, when they are ready.

The `epinions dataset` [1] was studied by Massa et al. [8] where also the interesting notion of "controversial users" was introduced in the context of the adversarial behaviour of some users.

The context of trust is extended by the user revision history in [14] and by the user's provenance in [3].

3 Proposed Experiments

As motivated at the beginning of the section 2, building on some recently published results, as explained above, we propose to experimentally test how much contextual information helps in trust prediction.

We propose the following experimental scheme:

1. Data: take a real dataset concerning trust network enhanced with some kind of contextual data such as ratings, user profiles, etc. At the beginning we plan to experiment with the publicly available `epinions dataset` [1], which contains both trust network and user recommendations, the latter one will be used as the contextual information
2. Sampling: from the dataset, take a random sample of user pairs for subsequent trust-prediction task. Due to the sizes of available real datasets, the epinions dataset in particular, sampling seems to be necessary, since repeating the experiment on *all* pairs of users in the dataset[3] seems to be computationally too expensive.
3. Training-testing split: as a common technique in supervised learning, to avoid evaluating the trust-prediction results on the same examples on which the predictor was trained (which is referred to as the *over-fitting* problem) split the sample into the *training* and *testing* subsets. We plan to tune the training/testing size ratio experimentally. The splitting should be *stratified* i.e. the proportion of the 'trusting' and 'distrusting' user pairs in sub-samples should be close to that observed in the whole sample.
4. Pure trust-topology prediction baseline: predict the value of trust (or distrust) between the nodes in the sample, based only on the trust-network information (*without* taking into account the actual known trust value on (u, v)). We propose to apply special trust-propagation techniques [4, 13].
5. Pure contextual-based prediction baseline: Compute contextual features, based on the additional information available in the dataset, such as user recommendations, similar to those described in [15, 2] and predict the trust using only these features
6. Contextually-enriched predictor: combine the trust-based and context-based features and finally train the combined trust predictor on the same sample as above.
7. Evaluation: compare the performance of the three prediction schemes above with use of some standard prediction-performance measures, such as prediction accuracy, the F-measure, etc.

The details about how the planned experiments relate to the previous research in the area and in which way the proposed experiments are novel are described in the following section.

[3] which is $O(n^2)$ for the number of users being n

4 Planned Contributions

The experimental work proposed in this paper is expected to contribute to the existing related research (see Section 2) in the following ways:

- supervised machine-learning approach, to the trust-prediction problem instead of a simple linear correlation measurement [15]. It seems that at least 2 different algorithms should be tested to have more reliable results, e.g. the Weka[4] implementation of the C4.5 decision tree and SVM.
- introduce some features based on especially designed trust-propagation models [4, 13]. Previously, only some general link-topology characteristics [7, 5] were used in this context.
- due to the extreme sparsity of the contextual data, when computing the ratings-based similarity between the users, we propose to first *cluster* the rated items with a method which is aware of the bi-partite structure of the rating graph. Independently, some dimension-reduction techniques such as SVD can be subsequently considered, if needed, to make the user profiles more overlapping. If a *hierarchical categorisation* of the rated items is available in the datasets, it should be used as well (as in [15]). Similar techniques concerning clustering the users could be considered. Due to this, the experiments can take into account also the users with little (or no) overlap in the rating-based profiles, in contrast to the previously reported approaches.
- in case such data is available, introduce a *distrust* prediction, which is a mathematically different task to trust prediction (e.g. distrust does not seem to allow for simple transitive propagation, in contrast to trust)
- when using rating data, take into account the true meaning of the values. Some previous approaches seem to not distinguish between 'positive' and 'negative' interpretation of the numerical values assigned to the ratings (and apply e.g. a dot-product similarity measure which is wrong in this context)
- on later stages of the research, if the textual ratings are available in the dataset, consider textual similarity of the textual reviews to compute more user similarity based features

To the best knowledge of the author, the combination of the propositions listed above is novel and can make the results of the proposed experimental research a potentially valuable contribution to the current research in the area.

5 Conclusions

Based on recently reported experimental results concerning observed significant correlation between user similarity and trust in social networks, a plan of further exploration of this direction is proposed here.

We propose a machine-learning approach to build a trust predictor and experimentally measure how much the contextual information can improve the

[4] http://www.cs.waikato.ac.nz/ml/weka

trust prediction accuracy compared to some trust prediction models which are based solely on pure topology of trust networks. We plan to start experimentation on the epinions dataset [1] and initially limit the contextual information to user recommendations, since such data is currently at our disposal.

References

1. `http://www.trustlet.org`.
2. Jennifer Golbeck. Trust and nuanced profile similarity in online social networks. Technical report.
3. Jennifer Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In *IPAW*, pages 101–108, 2006.
4. R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.
5. J. Karamon, Y. Matsuo, H. Yamamoto, and M. Ishizuka. Generating social network features for link-based classification. In *PKDD*, pages 127–139, 2007.
6. U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *AAAI*, pages 1377–1382, 2007.
7. David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of CIKM '03*, pages 556–559, NY, USA, 2003. ACM.
8. P. Massa and P. Avesani. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *AAAI*, pages 121–126, 2005.
9. Yutaka Matsuo and Hikaru Yamamoto. Measuring bidirectional effects on trust and rating on online social networks. Unpublished.
10. R. Neisse, M. Wegdam, P. Dockhorn Costa, and M. J. van Sinderen. Context-aware management domains. In B. Hulsebosch, G. Lenzini, and M. Wegdam, editors, *Proceedings of Context Awareness and Trust 2007 (CAT07), First International Workshop on Combining Context with Trust, Security and Privacy, Moncton, Canada*, volume 269 of *CEUR Workshop Proceedings*, pages 42–47, Nederlands, July 2007.
11. Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*, pages 77–92. Springer-Verlag, LNCS 3477/2005, May 2005.
12. Sini Ruohomaa and Lea Kutvonen. Making multi-dimensional trust decisions on inter-enterprise collaborations. In *Proceedings of the Third International Conference on Availability, Security and Reliability (ARES 2008)*, pages 873–880, Barcelona, Spain, March 2008. IEEE Computer Society.
13. Baoning Wu, Vinay Goel, and Brian D. Davison. Propagating trust and distrust to demote web spam. In *Workshop on Models of Trust for the Web*, Edinburgh, Scotland, May 2006.
14. Honglei Zeng, Maher Alhossaini, Li Ding, Richard Fikes, and Deborah L. McGuinness. Computing Trust from Revision History. In *Proceedings of the 2006 International Conference on Privacy, Security and Trust*, October 2006.
15. Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating interactions of trust and interest similarity. *Decis. Support Syst.*, 43(2):460–475, 2007.

Requirements Analysis for Identity Management in Ambient Environments: The HYDRA Approach

Hasan Akram, Mario Hoffmann

Fraunhofer Institute for Secure Information Technology,
Darmstadt, Germany
{hasan.akram;mario.hoffmann}@sit.fraunhofer.de

Abstract. The research field of Ambient Environments and Ubiquitous Computing aims toward the future vision of intelligent mobile and wireless network scenarios. In such environments where the wireless network consists of numerous nodes, like intelligent devices, sensors and mobile devices, a highly secured and well defined Identity (ID) Management System is required that deals with issues like virtual and temporary identities of users and devices as well as users' awareness in information disclosure and privacy. One major goal of the EU-project HYDRA[1] ("Networked Embedded System middleware for heterogeneous physical devices in a distributed architecture") is the support of developers of such ambient environments to manage context sensitive identity information and assure integration and interoperability of existing ID Management approaches. Based on this project in this paper we identify and analyze ten requirements for a middleware architecture to create a bridge between existing identity management technologies and also allow a framework to make them available for application developers of ambient environments.

Keywords: Identity Management, Ambient Environments, Security by Design, Privacy Protection, Identity Metasystem, HYDRA ID

1. Introduction

"7 trillion wireless devices serving 7 billion people in 2017" states the website[2] of the Wireless World Research Forum. This vision reflects the increasing trend of introducing micro- and nano-sized computers to everyday devices and tools (Ubiquitous Computing, Internet of Things). However, in such ambient environments not only computer systems become transparent and ubiquitous to users but also the users and their contexts become transparent and ubiquitous to the systems running in the background. And the more computers become transparent and ubiquitous the more the users' privacy and control is at stake.

---

[1] HYDRA: Networked embedded system middleware for heterogeneous physical devices in a distributed architecture. http://www.hydra.eu.com (2007) contract number: IST-2005-034891, duration: 07/2006-06/2010.

[2] WWRF: http://www.wireless-world-research.org

One necessary measure counteracting this rising challenge is a well defined combination of identity management for and virtualisation of both users and devices supported by future middlewares. This paper, therefore, focuses on a comprehensive requirements analysis for Identity Management in ambient environments and introduces recommendations for future system middleware architectures.

One of the main objectives of this paper is to draw a boundary of Identity Management support at middleware level in the context of HYDRA (Section 3). It is important to note that even a developer works keeping an end-user in mind. Therefore, as primary input for deriving middleware level requirements, the HYDRA home automation scenario is taken (Section 2), which basically illustrates application level use cases.

Taking advantage of the basic concepts of Kim Cameron's *Identity Metasystem* [1, 2], we show an extended application of the *Identity Metasystem* and derive ten particular requirements for Identity Management in ambient environments referring back to our HYDRA home automation scenario.

2. HYDRA Test Scenario: Intelligent Home

The HYDRA project uses the IDON method [6] for futuristic scenario definitions. By means of this systematic approach, fictitious scenarios have been derived in three domains - building automation, healthcare, and agriculture, which are likely to be practiced in reality in 2015 [3]. Many of these scenarios are derived from business cases from the perspective of an end-user, i.e. from application level. As a consequence Identity Management can have a large range of implications to information systems encompassing scenarios of access control, *Single Sign On (SSO)* in single and cross organizational domains, virtual identity, identity life cycle, session management and many other related issues. However, in case of designing a middleware for identity management the perspective of requirements analysis shifts from the end-user to a developer. This results in a different set of development time use cases from the very same application use cases.

With the intention to illustrate the necessity of an Identity Management System in HYDRA middleware we will take as a basis a detailed technical scenario of a heating system breakdown at "Krøyers Plads" housing complex located in Copenhagen that deploys the "Hydra Building Automation System" (HBAS) [5]. The resident living in a new flat in this building complex is equipped with automated lamps, computers and a wireless network, as well as a Hydra-enabled heating system and many other usual sets of automated devices. While the resident is at his office, the heating system of the flat breaks down and the water pressure rapidly decreases down to a level that is detected as an emergency situation by the HBAS which is shown as legend 1 in figure 1. As a result of that HBAS sends out an alert message to the resident (legend 2 in figure 1).

In order to get the heating system fixed as soon as possible the resident chooses a service provider from a list of providers matching the emergency requirements and his preferences best. The service provider then sends a service agent (e.g. a specialized technician) to the house. The challenge here is to allow a particularly

authorized service provider and his technician remotely to get into the house to fulfill a specific task. Therefore, included in the repair order a specifically restricted HBAS authorization ticket guarantees that in this case a service agent can enter the flat and get access to the heating system (legend 3 and 4 in figure 1). After entering the flat upon successful authentication procedure the service agent gets authorization to access additional context aware information required to perform his job (legend 4 in figure 1).

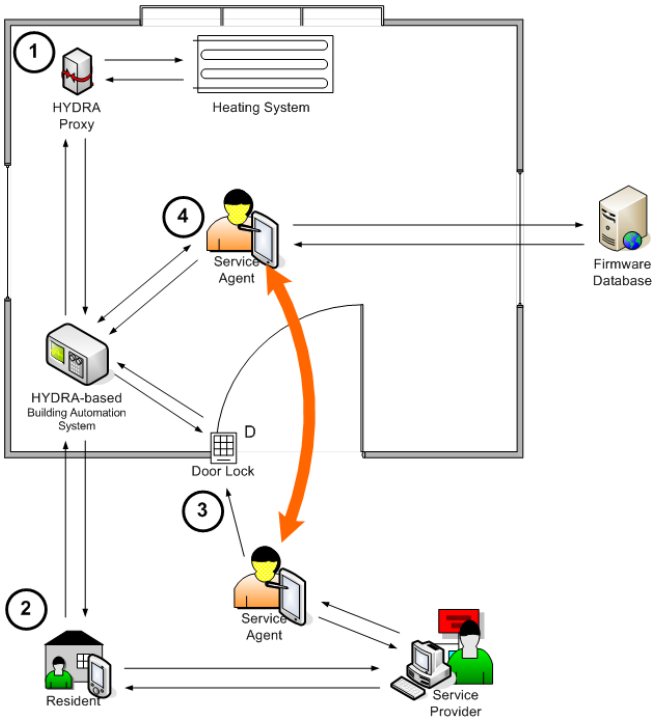

Fig. 1. Sequence of steps for the technical scenario [3].

This representative scenario can be basically adopted by many kinds of similar scenarios of remote authorization such as large housing areas with housekeeping service, office buildings with restricted access, airports, and hospitals. Thus, with the basic scenario of HYDRA being illustrated we can go one step forward in our process of HYDRA identity requirements analysis. In the next section we will provide a bird's eye overview of the *Identity Metasystem* and show how it relates to HYDRA use cases, which will be our basis on deriving HYDRA Identity Manager (HIM) requirements.

3. Application of Identity Metasystem in Extended Use Cases of the Hydra Scenario

This section is the basis for our requirement analysis process. The objective here is to establish the connection of *Identity Metasystem* and the given HYDRA scenario (Section 2). We start with the introduction of *Identity Metasystem* which is followed by an elaborate use case analysis focusing federated identity.

3.1 Identity Metasystem

Identity Metasystem [1, 2, 10, 11] is a claim[3] based architecture for an identity layer proposed by Kim Cameron in 2005 that uses federated identity as its underlying principle. The main goal of this architecture was to introduce an identity layer for the Internet that decouples Identity Management layer from the rest of the other layers in applications. Identity Metasystem is designed to be technology agnostic.

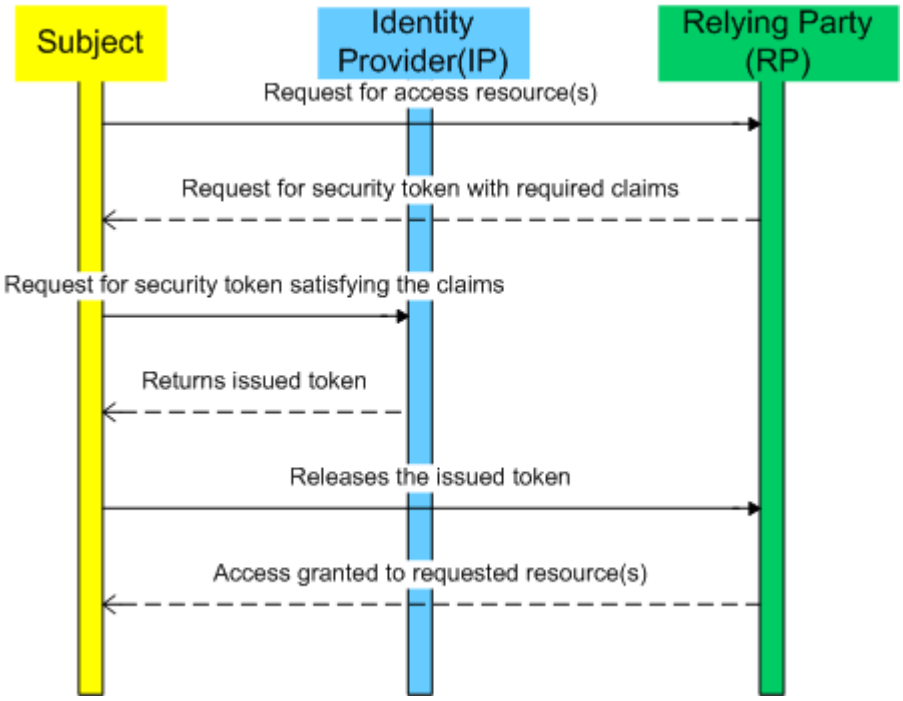

Fig. 2. Simplified sequence diagram of Identity Metasystem

Figure 2 illustrates a simplified version of the sequence diagram of Identity Metasystem. There are three roles in Identity Metasystem – the *Subject* (the user

[3] A claim is a declaration made by an entity. Examples include name, identity, key, group, privilege, and capability. - OASIS standard *Web Services Security: SOAP Message Security 1.0* [13].

whose identity is concerned), a *Relying Party*[4] (RP) and an *Identity Provider*[5] (IP). We can see in the sequence diagram that the Subject requests to have access to (a) particular resource(s) of a RP. The RP sends a set of claims or its identity requirements needed to access the requested resource(s) back to the *Subject*. The subject checks which IP(s) is suitable for this particular set of claims and chooses an IP. The *Subject* sends a security token request to the chosen IP. The IP issues and returns a security token satisfying the claims to the *Subject*. The *Subject* releases this token to the RP and gains access to his desired resource(s).

Based on this fundamental principle of Identity Metasystem, we will now analyze extended use cases of the Hydra in the next sub-section.

3.2 Extended Use Case Analysis of the Hydra Scenario

In this sub-section we will see use cases in HYDRA home automation scenario where the propagation of authentication information from entity to entity is based on contractual relationship. We will also observe how the three roles of Identity Metasystem explained in section 3.1 – *Subject*, *IP* and *RP* – shifts from endpoint to endpoint.

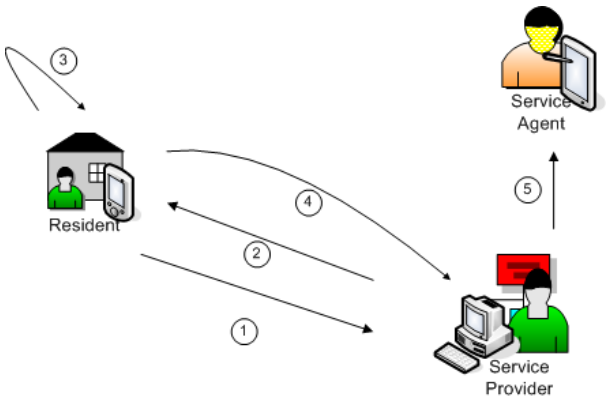

Fig. 3 Sequences in the process of the resident authenticating himself to the service provider and the service provider issues a cosigned token to the service agent.

Let us start with the use case shown in figure 3. This is the first identity federation in our scenario. In step 1 the resident is sending a request to the service provider for a service agent to be sent to his flat to fix his heating system. In step 2 the service provider is asking for his credential as a set of claims. Here the resident has an option to choose an IP that can satisfy the claims from the RP that happens to be the service provider in this case. For simplicity we assume that the resident himself is able to issue himself an identity token that satisfies the claims and would also be accepted by the RP. So, in step 3 the resident issues himself a token and in step 4 releases it to the service provider. After receiving this token the service provider issues a cosigned

[4] In federated Identity Management Relying Party is an entity that requests a digital identity of the user in form of a set of claims, issued from an Identity Provider.

[5] An Identity Provider is a trusted party that provides digital identities. It can be a third party or the user himself or even the Relying Party to whom the identity is to be disclosed.

token to a service agent (step 5) who is to be sent to the resident's flat for repairing the heating system.

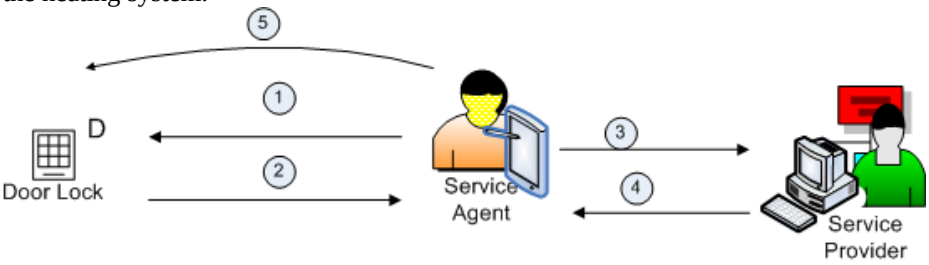

Fig. 4 Sequences in the process of the service agent getting authenticated by the door lock of the resident.

Now, let us look at another use case scenario shown in figure 4, where the service agent has to authenticate himself at the door lock of the resident's apartment. The roles - subject, RP and IP are shifted to the service agent, the door lock and the service provider correspondingly. Here, in step 1 the service agent sends a request to the door lock for accessing the flat. In step 2 the door lock sends a request for a security token as a set of claims. The service agent requests his IP (the service provider in this case) for a security token satisfying the claims. The service provider issues a token in step 4 and in step 5 the service agent releases this token to the door lock. Due to transitivity of authentication information flow as a part of the contract between the resident and the service provider, the door lock accepts his request; i.e. the door lock accepts authentication assertion (in form of a security token) from the resident, the resident sends the token to the service provider and the service provider issues a cosigned token to the service agent, consequently the door lock accepts the authentication information of the service agent. This process is repeated in each identity discovery taking place in the scenario.

Having these extended use cases being shown we can now derive specific *Identity Requirements* for the HYDRA middleware. Using this technical scenario and security requirements engineering of Hydra [9] we conceptualize the requirements of an *Identity Manager* being part of the Hydra middleware in the next section. Moreover, we identify and define what – from a *Security by Design* perspective – a developer may expect from Hydra middleware while developing an Identity Management System for ambient environments.

4. Identity Requirements in HYDRA

The illustration of the Hydra home automation scenario and the extended use case analysis with relation to Identity Metasystem make it obvious that developers of such ambient environment applications will have expectations up to a certain degree getting support for identity management processes by the middleware. Based on the given scenario, Hydra security, trust and privacy requirements as well as the trends toward identity requirements (e.g. Kim Cameron's Laws of Identity [1]) we define the following requirements for HIM (Hydra Identity Manager).

Definition: In Hydra an identity is assigned to any kind of entity, such as users, devices, and applications, being part of a Hydra-enabled infrastructure. An identity in Hydra typically comprises

(1) virtual temporary identifiers, e.g. specific Hydra IDs (HIDs) for (virtual) devices,

(2) an open list of specifying attributes, such as user preferences or device capabilities (e.g. stored in a device ontology), as well as

(3) a – if so timely restricted – history list of access events to and from this entity.

Sub-identities contain particular subsets of a Hydra identity depending on specific context information and collaboration partners. An entity may have different sub identities for and even in a specific context as long as a specific action still can be performed.

In a service oriented architecture Hydra's Identity Management System provides support to the developer to implement integrity, confidentiality and authenticity of such context specific actions, e.g. in work flows, transactions and processes performed by orchestrated services.

1. User Empowerment: Awareness and Control

The first identity requirement of HYDRA concerns the user in an ambient environment and emphasizes on two key words – "awareness" and "control". In a transaction taking place between two entities in HYDRA each entity must have full knowledge regarding the information he or she is about to disclose and to whom he or she is about to disclose. Besides having full knowledge about the information disclosure the entities must also have full range of control power to decide whether to disclose a particular set of information or not [1, 11].

An Identity Management System that does not confirm this need will suffer from serious security flaws. Lack of knowledge about the party to whom information is sent raises probability of phishing attacks. Information disclosed without knowledge of the user and lack of control of the user to decide what to disclose, violates his or her privacy.

2. Minimal Information Disclosure for a Constrained Use

Let us focus on our Hydra building automation scenario (Section 2) in order to clarify the second Hydra identity requirement. We have already stated that there is a contractual relationship between the resident and the service provider. Therefore, authentication information propagates in a transitive fashion to the service agent; i.e. since the agent is authenticated by the service provider, he is also authenticated by the resident and all the Hydra enabled devices in his or her apartment. In the process of fixing the heating system, the service agent will need to have access to certain information, e.g. the usage pattern of the heating system. The service agent will request the information needed in form of a set of claims. Here the service agent must be provided with a minimal information set that is only relevant for fixing the heating system. The usage pattern of the heating system supplied by the Hydra enabled

devices to the service agent must somehow guarantee that no other information is retrievable from it that goes beyond the necessity of fixing the heating system, e.g. the service agent should not be able to figure out from the usage pattern that during which period of the year the resident makes holiday or remains out of the flat.

3. Non-repudiation:

The term “Non-repudiation” has a traditional legal meaning and at the same times has a different meaning in terms of digital security [14]. We will focus on the latter meaning of “Non-repudiation” and then relate its necessity to our Hydra scenario (Section 2). In a crypto-technical sense transfer of data from one entity to another must guarantee authenticity, integrity and a time stamp, so that neither of the parties involved can deny that the transfer of the data took place.

Let us examine the issue of authenticity within the scope of the Hydra Scenario. The endpoint of the service provider receiving a message from the endpoint of the resident must know if the message is really transmitted from the resident or if it is under a spoofing or masquerade attack [5]. Therefore, there is a need of mechanism(s) that guarantees identity preservation.

To illustrate integrity, we continue with our running scenario example: the service provider receives a message from the resident over HTTP, he must guarantee the integrity of the message content. From a middleware viewpoint, there must be supports that allow the developer ensuring that the messages sent from one node to another is not being changed in an intermediary node or not under falsification attack [5]. To guarantee integrity it is also important that any kind of message manipulation has to be detected [15].

Another vital point is to make sure a time stamp is attached to the message. This is required to combat replay attacks. A time stamp attached to the message will make the message valid only for a certain period of time and as a result of that lower the probability of replay attacks.

Thus, we can sum up by saying that unforgeable identity, non-falsifiable message exchange, and provision of a time stamp is required in Hydra so that the identity of the sender and the integrity of the message cannot subsequently be refuted.

4. Support for directional identity topologies:

In the domain of ubiquitous computing communication takes place in various topologies and so does identity exchange. For example, when the resident communicates with the service provider (figure 3), it is a simple endpoint to endpoint (point to point or peer to peer) identity exchange. In the same scenario when the service provider comes into the flat of the resident, he or she needs to transmit his or her identity in a broadcast topology (figure 2) so that the Hydra enabled devices can detect his or her presence and take necessary actions. In the first example identity is unidirectional and in the latter case it is omni-directional. Based on this need of directional identity [1] the Hydra Identity Management middleware has to have supports for the following directional identity exchange topologies: 1. Broadcast (omni-directional) 2. Point to point (unidirectional) 3. Multicast (omni-directional and/or unidirectional).

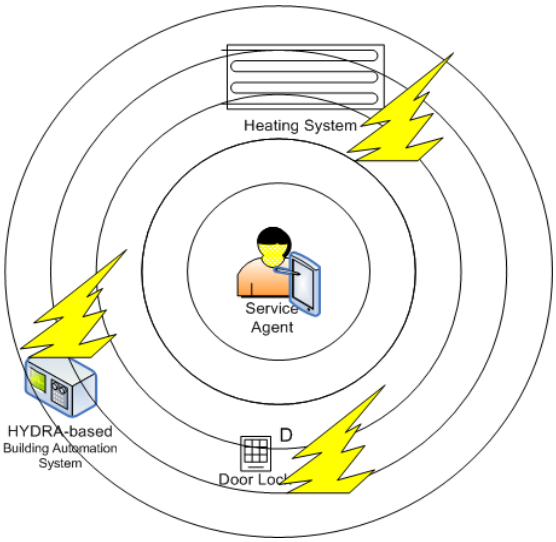

Fig. 5 The service agent transmits his or her identity in an omni-directional manner. The Hydra enabled devices senses his or her presence.

Fig. 6 Identity exchange taking place between the resident and the service provider. They both transmit their identity to each other in a unidirectional way.

Figure 5 and figure 6 illustrate omni-directional and unidirectional identities in the Hydra building automation scenario. In figure 5 the presence of the service agent is being sensed by the Hydra enabled intelligent devices at the resident's apartment, while the service agent transmits his identity in broadcast topology. In figure 6 the topology of unidirectional identity is shown in the use case where the resident and the service provider are exchanging their identities (point to point).

Following the notion of device discovery and service discovery in Hydra, we propose an identity discovery similar to figure 5 and 6. At Hydra middleware level device and service discovery will be transparent to a developer, he would rather be facilitated with identity discovery supports which is relevant for his identity layer of the application.

5. Universal Identity Bus:

Interoperability is one of the high level requirements of the Hydra project [4, 8]. Consequently, the Hydra Identity Management System inherently requires supporting interoperability between the garden varieties of Identity Management technologies available from different vendors. We propose a Universal Identity Bus (UIB) that will provide vendor to vendor interoperability functionalities. In order to achieve this

requirement the Hydra Identity Manager must support UIB that works as a bridge between different Identity Management technologies.

6. Provision of defining strength of identity:

In order to illustrate why Hydra necessitates provision of weak identities and strong identities, it is important to get back to the definition of identity in the context of Hydra shown in Section 2. We have seen that in Hydra identity relates to a person, a device or an application. Let us look at a case where a device is owned by a person. Here, the identity of the device is somewhat depended on the identity of the person, i.e. the identity of the device is incomplete without relating it to an identity of another entity. In a similar way many use cases may arrive where an identity does not suffice itself without being depending on an identity of another entity. Based on this criteria identity can be categorized to be strong (independent), weak (dependent) or somewhere in the middle. Thus, we can justify the requirement of a provision of having strength of an identity in the Hydra middleware. It is important to note that weak identities and strong identities are not the same as sub-identities (Section 2), which are basically subsets of identities. Identities or sub-identities both can be rated by their strength depending on their degree of being autonomous.

7. Decoupling identity management layer from application layer:

This requirement builds up another block on top of the "*Universal Identity Bus*" and separates the application layer from Hydra Identity Management layer. This is obligatory for the Hydra Identity Manager for two main reasons: 1) organizations are being able to change their identity policies without having an impact on the business layer and 2) the developers have an environment where they can work on the identity layer being transparent of the business layer or vice versa.

8. Usability issue concerning identity selection and disclosure:

We have already emphasized on the issue of empowerment of the user in case of revealing information in our first Hydra identity requirement. Lack of usability will make requirement 1 almost impossible to take place. In a user-centric design the user is the ultimate procurer and a methodic requirement specification of usability keeping the procurer in mind is unavoidable [12]. Therefore, HIM must facilitate the developer with adequate support for implementing usability.

9. Consistent experience across contexts:

Context is one of the major concerns in Hydra test scenario (Section 2) and identity and context are closely related. Therefore, while analyzing HIM requirements the issue of context is considered. In Hydra an entity and its identity will have an n to m relationship, i.e. one entity can have multiple identities and one identity can be possessed by several entities. For example, the resident has several identical sets of devices and he wants to use them with one single device identity. In this example one identity is shared by multiple entities. The example one entity having multiple identities would be, the resident has an identity at his work, a different one for his shopping web sites and another different one for heating system repairing service providers. In this n:m relationship of identities and entities it is very important to have consistence experience for the user depending on contexts.

Along with the consistencies among context, the identities provided in different contexts should also be independent of each other, i.e. the identity the user provides at work should not be related to his identity for his shopping website and vice versa.

10. Scalability:

In an ambient environment the number of nodes joining in and out is dynamic and thus the necessity of scalability in managing the identities of these numerous nodes is inevitable.

What does this at this point mean to a developer?
Let us look at the matter from a middleware point of view. A developer who will be working on an Identity Management System for similar scenarios defined in section 2 will have expectation from the middleware for supports so that he can ensure the requirements stated above. Our objective is to present the developer such an environment where the underlying technologies and other layers of the Hydra middleware are transparent to the Identity Management layer and the developer is able to totally focus on his Identity Management related needs.

6 Comparison with related works

Identity Management in pervasive computing has been explored by researchers since almost the very beginning of pervasive computing. Requirements and principles of Identity Management has been analyzed and derived based on certain needs in certain scenarios. Obviously, these related works have some commonalities and disparities among themselves. In this section we briefly report a comparative study of our work with respect to a few selected related works in Identity Management in ubiquitous computing. In this comparative study we also highlight a justification of our proposal of requirements rather than choosing one of the existing works.

There are related works where they deal with application level requirements analysis for identity preservation in pervasive computing, e.g. the requirements proposed by Roy Champbell [17]. In their paper they have shown requirements of security in such ambient scenarios. Although there are partial overlaps with our requirements, there is a fundamental difference of looking at the problem from application perspective and a middleware perspective.

Privacy principles described by Langheinrich [18] also have some overlaps and as well as differences compared to our work. The differences and similarities are due to the fact that identity is a broader concept and it comprises many other elements including privacy.

Jendricke [16] proposed context driven Identity Management to comply with principles of Langheinrich [18] and illustrated their architecture and prototype. Again, it is designed from an application viewpoint. Moreover, the solution proposed in this paper is not federation driven. We have already seen in the extended use cases (Section 3.2) that Hydra scenario is federation driven. Therefore, this solution was also not totally pluggable to our need.

Kim Cameron's [1] laws of identity is also not fully meant for ambient environment and focused on the present situation of internet, whereas Hydra scenario

is focused on a projected scenario in 2015. Therefore, the seven laws stated in his white paper do not totally suffice our needs as well. However, one fundamental principle that is common with our scenario is the concept of federation. Identity metasystem [2, 9] architected by Kim Cameron is a federation based concept and is very much applicable to Hydra scenarios. This motivation led us to apply the concept of identity metasystem in our use case analysis (Section 3.2) and derive HIM requirements based on the use case analysis.

Finally, I would like to sum up by saying that if a Venn diagram is constructed for all the sets of laws of Identity Management proposed so far, there will always be an intersection region. At the same time there can be regions in each of these sets which are non-overlapping. This is simply because all these laws are based on some variable parameters; namely - perspective, time, computing environment etc.

7 Conclusion & Outlook

In this paper we have illustrated the requirements analysis of Identity Management in futuristic scenarios of ambient environment or ubiquitous computing from a middleware viewpoint. We have shown the significance of Identity Metasystem [1, 2, 10, 11] in such futuristic scenarios and also seen the propagation of authentication in federated Identity Management. The following list summarises the requirements identified and analysed:

1. User Empowerment: Awareness and Control
2. Minimal Information Disclosure for a Constrained Use
3. Non-repudiation
4. Support for directional identity topologies
5. Universal Identity Bus
6. Provision of defining strength of identity
7. Decoupling identity management layer from application layer
8. Usability issue concerning identity selection and disclosure
9. Consistent experience across contexts
10. Scalability

The future goal of this research work is an evaluation of the state of the art technologies that best suites the requirements and eventually derive an architecture based on the requirements analysis of this paper. The results of the architecture specification for ambient environments will be published soon. Choosing the best suited technology for Identity Management, we plan to build the Hydra Identity Management SDK as a set of service library and integrate it into the HYDRA middleware. This part will be published by the end of this year.

Acknowledgements

We would sincerely like to thank Julian Schütte (Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany) for his review and helpful remarks on this paper, which certainly added value to the entire work.

References

1. Cameron, K, Laws of Identity (2005), Microsoft Corporation.
2. McLaughlin, L., What Microsoft's identity metasystem means to developers, *Software, IEEE* , vol.23, no.1, pp. 108-111, Jan.-Feb. 2006.
3. HYDRA, Deliverable D3.3 Draft of architectural design specification, 11 May 2007, Version 1.3.
4. HYDRA, Deliverable D7.1 Security Requirements Specification, 8 March 2007, Version 1.0.
5. HYDRA, Deliverable D2.1a Scenarios for usage of Hydra in Building Automation, 25 January 2007 - version 1.41.
6. Galt, Chicoine-Piper, and Hodgson (1997). IDON Scenario Thinking: How to Navigate the Uncertainties of Unknown Futures. IDON Ltd.
7. The Hydra Project, http://www.hydra.eu.com
8. Hoffmann, M., Badii, A, Engberg, S., Nair, R., Thiemert, D., Mattheß, M., Schütte, J., "Towards Semantic Resolution of Security in Ambient Environments", *Ami.d – 2nd Conference for Ambient Intelligence Developments*, September 2007
9. Cameron, K., Jones M. B., Design Rationale behind the Identity Metasystem Architecture, http://www.identityblog.com/, http://research.microsoft.com/~mbj
10. J. Miller, Yadis 1.0, http://yadis.org/papers/yadisv1.0.pdf, March 2006
11. Bertocci, V., Serack, G., Baker, C., Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities, December 27 2007, Addison-Wesley.
12. Artman, H. 2002. Procurer usability requirements: negotiations in contract development. In *Proceedings of the Second Nordic Conference on Human-Computer interaction* (Aarhus, Denmark, October 19 - 23, 2002). NordiCHI '02, vol. 31. ACM, New York, NY, 61-70. DOI= http://doi.acm.org/10.1145/572020.572029
13. *WS-Trust 1.3*, OASIS Standard 19 March 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html#_Toc162064937
14. McCullagh, A., Caelli, W., Non-Repudiation in the Digital Environment, *First Monday, volume 5, number 8* (August 2000), URL: http://firstmonday.org/issues/issue5_8/mccullagh/index.html
15. Müller, G., Rannenberg, K., Multilateral Security in Communications, Vol.3, Technology, Infrastructure, Economy**,** Addison-Wesley, 15. July 1999.
16. Jendricke, U., Kreutzer, M., Zugenmaier, A., 2002. Pervasive privacy with identity management, *In Proceedings of the Workshop on Security in Ubiquitous Computing, UbiComp* 2002, Sweden.
17. Campbell, R, Al-Muhtadi, J., Naldurg, P., Sampemane, G., and Mickunas, M. D. Towards security and privacy for pervasive computing. *In Proceedings of International Symposium on Software Security*, Tokyo, Japan, 2002.
18. Langheinrich, M., Privacy by Design – Principle of Privacy-Aware Ubiquitous Systems, *Proceedings of the UBICOMP* 2001.

Persistent Authentication in Smart Environments

Mads Syska Hansen, Martin Kirschmeyer, and Christian D. Jensen

Department of Informatics & Mathematical Modelling
Technical University of Denmark
Christian.Jensen@imm.dtu.dk

Abstract. Inhabitants in smart environments are often authenticated when they enter the smart environment, e.g., through biometrics or smart-/swipe-card systems. It may sometimes be necessary to re-authenticate when an inhabitant wishes to enter a restricted area or access ambient services or location based information, e.g., it is common to have swipe card terminals placed next to doors to restricted areas. This means that all access to protected resources must have individual means of authenticating users, which makes the access control system more expensive and less flexible, because access controls will not be installed unless it is absolutely necessary. The cost of installing and maintaining an authentication infrastructure and the inconvenience of repeatedly authenticating toward different location based service providers mean that new models of authentication are needed in smart environments.
This paper defines a persistent authentication model for a smart environment, which tracks inhabitants in the smart environment from the point of authentication to the protected resource, thus rendering authentication persistent by correlating the initial authentication event with the access control request. We present a proof-of-concept implementation of the proposed mechanism, which employs camera based tracking with a single stationary 3D camera that uses the "time of flight" principle. A preliminary evaluation of the proposed mechanism indicates that persistent authentication is technically possible with the proposed hardware. The proposed model is sufficiently general to allow the addition of more cameras or supplemental tracking technologies, which will improve the robustness and scalability of the proposed mechanism.

1 Introduction

Smart environments may be defined as *"a small world where all kinds of smart devices are continuously working to make inhabitants' lives more comfortable"* [1, p. 3]. It is generally assumed that a number of sensors are embedded in the environment to determine the current context of the inhabitants, so that the underlying system can anticipate their needs and provide them with services that facilitate their everyday life. Ideally, the provision of these services should be completely transparent to the inhabitant, who simply observes that services are available as they are needed, e.g., front doors open when they approach or lights are dimmed in the living room when the home cinema system starts.

Provision of such context-aware services in a smart environment requires knowledge about the inhabitants that are present in the environment and their current context.

Information about the inhabitants may include their identity, service history or profile, while information about the context includes the location of the inhabitants – either their location in absolute coordinates or their location relative to other inhabitants and the points of service provision. Moreover, knowing the exact location of inhabitants may help the smart environment to focus on the sensors that cover the areas where people are present. Fusion of data from diverse sensors and tracking of inhabitants' locations and behaviour allows the system to build accurate profiles of preferences and interaction behaviour. This raises important questions about the protection of the collected information and the privacy of inhabitants, which we do not address in this paper. Detailed knowledge of the environment, however, may also be used to enhance existing security services and provide stronger or more convenient security mechanisms; this is the topic of this paper.

In this paper, we examine the problem of authenticating principals[1] in smart environments. We propose an authentication model called *Persistent Authentication In Smart Environments* (*PAISE*), which combines traditional authentication mechanisms with sensing technologies and tracking capabilities offered by the smart environment. Without loss of generality, we limit our discussion to a single application of a secure service in a smart environment, namely an access control mechanism that controls the lock on a door to a restricted area. This application has all the essential properties of a secure server, but its familiarity and simplicity facilitates the discussion of our model.

There are different ways to ensure that the person who was authenticated is also the person who is trying to enter the restricted area. The simplest and most secure solution would probably be to enforce that only one person at a time is able to enter the corridors between the point of authentication and the restricted area, but such a solution is obviously too restrictive. Another solution is to track inhabitants from the point of authentication to the restricted area, thus correlating the authentication event with the access control request; this is the approach taken in the *PAISE* model.

The *PAISE* model proposed in this paper has been implemented in a simple prototype, which uses camera-based tracking using a 3D camera. Our evaluation shows that a single TOF camera is sufficient to track a small set of individual users in many situations, but that further work is required to improve the persistence, robustness and scalability of the system. We do, however, believe that multiple calibrated cameras [2] may help address these issues.

The rest of this paper is organised in the following way. Section 2 provides motivations for the model and a brief analysis of authentication in smart environments is presented in Section 3. Persistent authentication and the *PAISE* model is presented in Section 4 and a brief overview of the design and implementation of our *PAISE* prototype is presented in Section 5. The preliminary evaluation of our prototype is presented in Section 6 and related work is examined in Section 7. Finally, Section 8 presents our conclusions and outlines some directions for future work.

[1] We generally use *inhabitant* to refer to people in general and *principal* when we consider people in a security context.

2 Motivation

Physical access control is traditionally based on the authenticated identity of the principal. The authentication could be performed by a human guard who knows the user, by using a key, a swipe-/smart-card, a personal identification number (PIN) or a combination of the above. Many buildings have zones with different access control restrictions, so a principal moving between different zones would need to authenticate repeatedly, which would be considered an inconvenience and a distraction by most people. As a small example, consider the entrance to Building 322 at the Technical University of Denmark shown in Figure 1. Swipe-card access control and different restrictions for general access to the building and access to each of the hallways, means that principals need to authenticate twice to enter the hallway on the ground floor, even though there are no more than 6 meters between the two points of authentication. With swipe-card access to individual offices, staff would have to authenticate a third time before they can enter their own office.

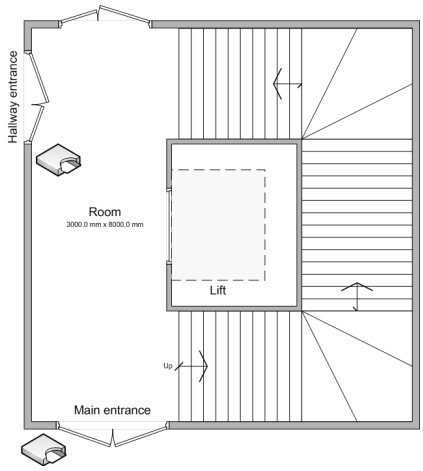

Fig. 1. East entrance to Building 322 at the Technical University of Denmark.

Depending on the physical access control policy implemented, a building could hold many points of authentication, which introduces the common known trade-off between security and usability. The more secure a solution is the less user-friendly it tends to be.

To improve the usability of such an access controlled environment one could try to introduce a system using only a single point of authentication. This solution would, however, introduce a new problem, which corresponds to what is commonly known in software as the time-of-check-to-time-of-use (TOCTTOU) problem; a kind of software bug that can be explained as a race condition between the check of the security credentials and the use of that checked credential. In a physical access control system, this problem can be translated to a location-of-check-to-location-of-use (LOCTLOU)

problem. The problem is still a race condition between the point where the principals verifies his identity and the intended use of that verification. If the principal is not alone between the verification and the point of use, then other inhabitants could usurp the authentication and enter restricted areas, which they were not authorised to enter - a race condition between users in the system.

The simplest way to protect against LOCTLOU, would be to restrict the area between the authentication and the intended use to one person at a time. This solution would, however, not be suitable, because it would impose too many restrictions on the simultaneous movements of people in the building.

3 Authentication in Smart Environments

It is a general requirement in smart environments that services are only provided to authorised users, e.g., the front door should not open for everybody. This means that services in a smart environment need to authenticate users in order to determine whether a principal is authorised or not. Authentication, however, is normally a process that requires active participation by the principal, e.g., presenting a badge, entering a password, swiping a finger across a biometric reader, etc. If we are to implement Mark Weiser's vision of ubiquitous computing [3], the authentication technologies employed in a smart environment need to be "calm" [4], which means that they should require minimal attention from the principals.

There are essentially two ways to implement calm authentication: either the principals are continuously authenticated in a way that they do not notice or they authenticate in a few strategic locations and the smart environment tracks the principal and makes the authentication information available to services as they are required.

In the first case, the authentication may either be based on biometrics that can be measured from a distance or the principal can be required to carry a small authentication token with short range communication capabilities that authenticates the principal toward the context-aware service providers in the smart environment. Typical biometric authentication technologies include fingerprint recognition, iris-/retina-scan, voice recognition and face recognition. Face recognition is the only of these technologies that does not require user involvement, but there are generally serious problems with false positives and false negatives, so we do not believe that the technology is sufficiently mature and secure for our scenario. It is also important to note that the failure mode of biometric authentication is absolute: false positives mean that unauthorised principals are granted access to a resource and false negatives may well imply that the principal has changed appearance and has to enroll with the biometric authentication system again. Smart wireless authentication tokens are convenient in many ways and, if properly used, the authentication results are both secure and non-intrusive. However, they do introduce problems when the authentication tokens are forgotten, lost or stolen. In some of these cases, principals will be tempted to borrow authentication tokens from each other, which leads to erroneous authentication.

In the second case, existing authentication technologies are used, so that authentication terminals are located in a few strategic places in the environment; we call these locations the *point of authentication*. Sensors in the smart environment are then used to

track the principals from the point of authentication, so authentication is only done once when the principal enters the smart environment from the outside. The authentication system associates the result of this authentication with the principal as he moves around in the smart environment, thus rendering the authentication persistent.

4 Persistent Authentication

The idea behind persistent authentication is to replace repetitive re-authentications with a system that tracks inhabitants in a smart environment from the point where authentication is done to the point where access control is enforced, i.e., it translates authentication in time and space from where it is done to where it is needed. This means that the event of authentication *"sticks"* to the principal, thus making it persistent.

4.1 *PAISE* Model Overview

The *PAISE* model defines four major components in a persistent authentication system: an authentication system, which is able to authenticate principals; a smart environment, which delivers the sensor data needed for tracking; an access control mechanism, which acts on the result of persistent authentication and the core component of *PAISE*, which combines the information from the authentication system and the smart environment, tracks authenticated principals in the smart environment and forward the necessary data to the access control mechanism. These components are shown in Figure 2.

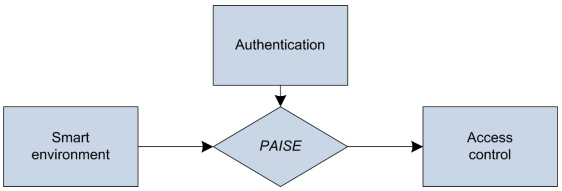

Fig. 2. The idea is to combine information from an initial authentication with information from a smart environment to perform persistent authentication.

In addition to these four components, *PAISE* also defines authentication zones and authorisation zones in the smart environment. An *authentication zone* defines the area in front of the authentication mechanism which is large enough to hold a single principal. The smart environment delivers a constant stream of sensor data to the core component, but tracking is only initiated when a principal has entered the authentication zone and successfully authenticated himself. The authentication zone must be small enough to ensure that the authentication event can be reliably linked to the principal. A typical authentication zone, in a smart environment, would be an area 0.5m $\times$ 0.5m in front of a swipe-card terminal. An *authorisation zone* defines the area in which the access control policy of a location based service must be enforced. When new principals enter an

authorisation zone the persistent authentication is forwarded to the access control mechanism of the location based service provider, which is then able to determine whether access should be granted. In the case of access through a door, in a smart environment, the authorisation zone must be small enough to ensure that most principals are able to reach and open the door while it is unlocked, but also large enough to ensure that nobody outside the authorisation zone is able to pass through the door while it is open. This allows the system to enforce the constraint that the door can only be unlocked if there are no unauthenticated or unauthorised principals inside the authorisation zone, thus preventing tailgating.

4.2 *PAISE* Security

The basic authentication in *PAISE*is performed by an authentication system that is external to the model. This means that the model supports *state of the art* authentication mechanisms based on passwords, PIN, smart-cards, authentication tokens, biometrics or multi-factor authentication [5]. The security of persistent authentication is therefore primarily a question of the systems ability to track principals after authentication.

There are different ways to locate or track inhabitants in a smart environment. The most common methods are: motion detectors based on photocells, infrared light or lasers; acoustic detectors similar to sonars or based on triangulation with multiple microphones; camera-based location and tracking systems; pressure sensitive floors [6] and token-based location and tracking systems, such as the Active Badge system [7], where each principal wears an active authentication token used to determine their location and track their movements in the smart environment.

In order to determine the overall security of a *PAISE* implementation, it is important to evaluate the tracking mechanism with respect to persistence, robustness and scalability, which we define in the following.

Persistence: The ability to track the principal and maintain the authentication. Persistence primarily address problems that arise in the day-to-day operation of the system, e.g., tracking may be lost if sensors are temporarily blinded by a flash from a tourist's camera.

Robustness: The ability to resist malicious and colluding principals' attempts to usurp the identity of other principals (each other in the case of colluding principals).

Scalability: The ability of the authenticate a large number of principals in a potentially large environment.

The different location and tracking technologies have different properties with respect to the accuracy; simplicity of installation and maintenance; and cost of installation and operation, but none of them are perfect. It is therefore important to force the system to a fail safe state, i.e., immediately classify the principal as unknown, when the tracking is lost or there is a risk of mistaken identity. As authentication always precedes authorisation, this means that no principal will ever be authorised based on suspect authentication information. If the authentication is lost, the principal has to re-authenticate at the nearest authentication zone, but this will be a rare event if the persistence and robustness of the tracking mechanism is high. Moreover, it is possible to place additional

authentication zones at several, strategically selected, locations in the smart environment so that principals will never have to move far if they need to re-authenticate. The number and locations of such additional authentication zones depend on both the persistence and robustness of the tracking mechanism and on the topography and the movement patterns of principals in the smart environment.

5 *PAISE* Prototype

In the following, we present a brief overview of the *PAISE* prototype that we have developed.

5.1 Overview

An implementation of *PAISE* consist of the four components shown on Figure 2, which is translated into a smart environment as illustrated on Figure 3.

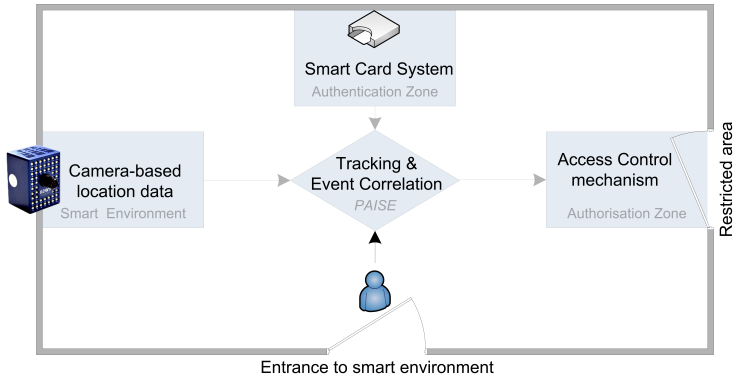

Fig. 3. Overview of the *PAISE* subcomponent connections.

The Figure shows a smart environment consisting of a single room, which has a camera-based location system. The room has access from the outside and provides principals, who authenticate using a smart-card based system, access to a restricted area. The decision to use a smart-card authentication system introduces many of the problems of token-based authentication and location systems to our prototype, i.e., that tokens can be forgotten, borrowed, lost or stolen. We would like to remind the reader that the choice of authentication system is external to our model and we simply chose smart-cards because it is reasonably secure and the hardware was available. Replacing smart-card based authentication with a system based on passwords and/or biometrics will completely eliminate this problem from our implementation. We describe each of the other elements of our prototype in greater details in the following.

5.2 Smart Environment

The sensors in the smart environment in our prototype consist of a single MESA Swiss-Ranger SR-3000 camera, which operates on the Time-Of-Flight (TOF) principle. The camera uses near-infrared[2] LED's (wavelength 850 nm[3]) to generate a depth image based on the Time-of-Flight principle. Light is sent out and the camera calculates the distance d_O based on the amount of time it takes the light moving to the object and back to the camera.

$$d_O = \frac{c}{2f} \cdot \frac{\varepsilon}{2\pi}, \tag{1}$$

where f is the frequency, c is the speed of light and ε is the phase [8, p. 9-16].

The TOF camera is able to deliver depth information out-of-the-box as the hardware inside the camera makes the needed calculations. The prototype is however quite expensive (approximately 5,000 Euros) - but the manufacturer of the Swiss-Ranger TOF camera (MESA) states that the camera should have a price in the same range as a normal web camera when a mass production starts.

5.3 Tracking in *PAISE*

Based on the depth information provided by the TOF camera, the *PAISE* prototype is able to identify objects that have the same distance and direction from the camera; such objects are commonly referred to as blobs. Each blob is a representation of an object, which is projected on to the floor of a virtual room[4] and tracking is done in two dimensions. Further details about how blobs are constructed and tracked is published elsewhere [9].

The interaction between the physical and virtual world is an important factor in *PAISE*. Decisions such as whether a user is granted access to a specific area is an access control decision, which need to be made by the system based on the location and the clearance of the user.

The main idea is to position the principals (by the location given by the tracking) in the virtual room, which corresponds to the real physical room. The locations of the principals should be checked against the predefined zones and if the users are located in these, an appropriate action is taken according to the decision tree illustrated in Figure 4. The current prototype implements the following security policy:

Authentication: The oldest blob at the authentication zone will get the clearance present at the authentication server. This means that if no credentials are present then the blob will remain unauthenticated.

Clearance: The authentication is used to label an clearance on a blob. If the blob is lost/eliminated the clearance is eliminated with the blob.

[2] Near-infrared light has a wavelength between 700 nm and 2000 nm, and is used for night-vision devises. Source: Britannica Online article 9002311 [February 13, 2008]

[3] Source: http://www.mesa-imaging.ch/prodviews.php [February 13, 2008]

[4] The virtual room is represented by an image of dimensions fitting the real room.

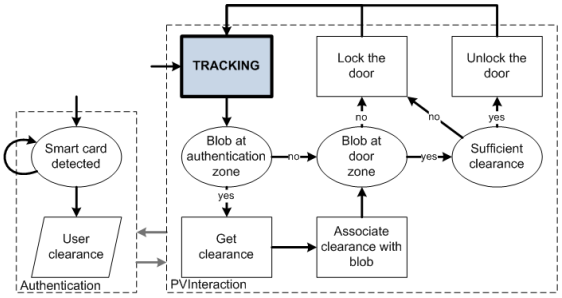

Fig. 4. Decision tree describing the access control of the electronic door lock.

Door: Access to the restricted area behind the electronically locked door is granted if a blob with the correct clearance is present at the zone (door). If more than one blob is present access is granted if just one of the blobs has the correct clearance. This allows authorised principals to accompany guests around the facility.

This security policy is quite simple, but it suffices to illustrate the advantage of persistent authentication in smart environments. The modular design of the *PAISE* prototype means that it is fairly easy to replace the authentication mechanism (*Authentication*) and the access control policy and mechanism (*Clearance* and *Door*) to reflect the needs of real environments. As an example, it is possible to implement an authentication mechanism, which does not authenticate principals if more than one blob is present in, or near, the authentication zone. It is also possible to enforce other access control policies, such as an access control policy that denies access to the protected resources when an unauthorised principal is present in, or near, the authorisation zone. This would limit the possibility of tailgating, social engineering and coercing an authorised principal to give access to an unauthorised principal.

The design and implementation of our prototype is described in greater detail in the M.Sc. Thesis of Kirschmeyer & Hansen [10].[5]

6 Evaluation

The evaluation of our prototype must determine whether it meets the important security properties in persistent authentication, namely persistence, robustness and scalability.

6.1 Evaluation Setup

The evaluation was performed in a deserted hallway in our building. The 3D camera was mounted in an angled top-view position just below the ceiling at one end of the hallway. The angle between the ceiling and the top of the camera was as small as possible. This

[5] This report is currently not available online, but it can be obtained from the library of the Department of Informatics and Mathematical Modelling at the Technical University of Denmark.

resulted in a scene that was 6.4m long, 2.3m wide and 2.3 m high. The scene was illuminated by fluorescent lights mounted in the ceiling.

The core component of the *PAISE* prototype runs on an IBM ThinkPad T60, with a 1.8GHz processor, 1.5GB internal memory and plenty of available disk space. We used a simple smart-card based authentication system, but since this is an exchangeable part of the system, and external to the *PAISE* model, we consider it beyond the scope of this evaluation.

6.2 Persistence

A number of experiments were designed to test the prototype's ability to track principals under different conditions. In particular, we wish to test how the system deals with: multiple principals, foreign objects, velocity of principals, partial occlusion and close contact between principals. The results of our evaluation is shown in Table 1.

Test	Scenario	Expected outcome	Result
1	Two principals walk in the same direction	The tracking should identify and follow the two separate users	success
2	Two principals walk in opposite directions	The tracking should identify and follow the two separate users	success
3	Two principals cross their path	The tracking should identify and follow the two separate users	success
4	One principal leaves a small object (a bag) in the scene	The tracking should follow the user and ignore the small object	success
5	One principal leaves a large object (a ladder) in the scene	The tracking should follow the user and ignore the large object	success
6	Two Principals talking (one principal is partially occluded)	The tracking should identify and follow the two separate users	success
7	Two principals shaking hands	The tracking should identify and follow the two separate users	success

Table 1. The results of the persistence evaluation.

The first three experiments validate that tracking works under normal circumstances and experiments 4 and 5 show that the system is able to handle foreign objects. Experiments 6 and 7 demonstrate the advantage of using a TOF camera, because the two principals have different distance to the TOF camera.

6.3 Robustness

An evaluation of the robustness must assess the persistence of the authentication with respect to properties that are open to manipulation by a malicious attacker. Such properties include lighting, properties of clothing, posture and velocity of principals in the smart environment.

The TOF camera operates with near infrared light, so the changes in the visible spectrum of light in experiment 1 has no effect. The attempt to *blind* the TOF camera

Test	Scenario	Expected outcome	Result
1	One principal walks in light/darkness	The tracking should follow the user regardless of the illumination of the scene	success
2	One principal points a near infrared light source at the TOF camera	The tracking system will be blinded (denial of service)	partial success
3	One principal changes clothes from black to white	The tracking should follow the user regardless of the change in black/withe contrast	success
4	One principal walks wrapped in tinfoil	The tracking should follow the user despite reflections	success
5	One principal stands and sits down	The tracking should follow the user despite the change in posture	success
6	One principal running	The tracking should follow the user	failure
7	One principal jumping around	The tracking should follow the user	failure
8	Usurpation attempted (two principals, but only one authentication)	The usurping principal will not be authenticated when he enters the authorisation zone	success
9	Two principals bump into each other	The tracking should identify and follow the two separate users	partial success

Table 2. The results of the robustness evaluation.

in experiment 2 is only partially successful, because the attacker has to be very close to the TOF camera (approximately 1m) before the attack is effective. The TOF camera has known problems with measuring distance to objects that have sharp contrasts between black and white. The attacker in experiment 3, attempts to exploit this problem by wearing a black jacket and white trousers; he further takes off the jacket to reveal a white t-shirt underneath, but the *PAISE* tracking system was able to persistently track the principal. Neither the reflections created by the tinfoil in experiment 4, nor the change in posture in experiment 5 has any effect on the tracking. Experiments 6 and 7 show that the current tracking system is unable to manage principals who move very quickly or who frequently change direction and velocity. The heavy computational load means that the frame rate of the tracking system is unable to handle large variations in the scene. In experiment 8, one principal is waiting for the other principal to authenticate and races to the authorisation zone ahead of him, but the access control mechanism does not grant access. Experiment 9 is a partial success; tracking is temporarily lost at the moment the two principals collide, but the system correctly identifies and tracks the two principals after the collision. Further experimentations are needed to build confidence in the system's ability to correctly identify principals after collisions.

The robustness of the tracking system, using a single TOF camera, is surprisingly high. Moreover, we believe that the robustness can be improved significantly by using a multi-modal tracking mechanism, e.g., a simple web camera could improve tracking robustness by including information about the colour of clothes.

6.4 Scalability

We did not design specific tests to determine the scalability of our current prototype. Our general experience with the system, however, indicate that the scalability is unsatisfactory. With more than a handful of people on the scene, the frame rate that the system is able to process with the current hardware drops below the level necessary to provide reliable tracking. However, the algorithms used to track each individual person has no interaction with the tracking of other people, so the task is well suited for parallel computation. We therefore expect to be able to develop a more scalable version of our prototype, where parallel processors are used to track principals in the scene.

7 Related Work

Corner and Noble [11–13] examine the problem of authentication when mobile devices are lost or users leave a work station logged in. They define traditional authentication mechanisms as *persistent* because they rarely limit the duration that the authentication is valid, so a user may leave a computer logged in for several days without a screensaver. This means that anyone who steals a device that is logged in or gets physical access to the workstation may usurp the authentication of the original user.

They define an *transient authentication* mechanism, where all data in the system is encrypted and a small *authentication token*, worn by the user, is needed to provide access to the encrypted data, thus ensuring that access can only be granted when the token is in close proximity to the system where the user is logged in. The token stores the cryptographic keys and the proximity mechanism is based on short range wireless communication.

The definitions of persistent and transient authentication by Corner and Noble are device centric, authentication sticks to the device as long as the user is present, so restrictions may be put on the users, e.g., they have to wear the authentication token. This creates problems when authentication tokens are forgotten, borrowed or lost. Our definition of persistent authentication is user centric, which means that authentication sticks to the user as long as the tracking from the last authentication zone is considered reliable. This means that any authentication mechanism, e.g., passwords, PIN or biometrics, can be used and that no additional requirements are placed on the user.

Bardram et al. [14] defines a context-aware user authentication mechanism, where users need a smart card to identify themselves to the system and an RFID based tracking system is used to authenticate the user. This adds complexity for the user, by requiring that he remembers two tokens, without offering significantly improved convenience, i.e., the user still has to insert the smart card into the system whenever authentication is required. Our proposal removes the need to perform specific authentication actions as long as the tracking is considered reliable.

Klosterman and Ganger [15] define a *continuous biometric-enhanced authentication* mechanism, which uses a biometric authentication module, based on face recognition, to periodically re-authenticate users who are logged in to the system. If, at some point, the biometrics of the user sitting in front of the monitor does not correspond to the biometrics of the authenticated user, re-authentication is required. This means that continuous authentication is achieved without addition requirements are placed on the user,

but their system authenticate a specific user at a specific location, where we propose to track the user so that his authentication may be reused in different locations.

8 Conclusions

In this paper we examined the problem of user authentication in smart environments. We proposed a persistent authentication model, which tracks principals in the smart environment and binds authentication information (a clearance) to principals whenever they authenticate with the system. This means that mobile users are transparently authenticated toward location based services in the smart environment. This has obvious privacy implications which we aim to address in future work.

We presented a brief overview of the prototype implementation of *PAISE*, which we have developed at the Technical University of Denmark. We have conducted a series of different experiments to evaluate our prototype and some of the results of these experiments are presented in the paper.

The evaluation shows that the *PAISE* prototype is able to track a small number of simultaneous principals who move normally in a smart environment, so that once a principal has been authenticated, the result of this authentication can be associated with the principal as he moves around in the smart environment. Our evaluation also demonstrate that the current prototype is unable to track principals who move very fast or who changes direction or location very quickly. We believe that this is primarily caused by the limited computational resources available for the prototype, which results in a relatively low frame rate from the tracking subsystem (see also the discussion of scalability below). We have also identified problems when principals are in very close contact with each other, e.g., two principals hugging. We conjecture that both of these problems may be addressed by adding more sensors to the smart environment, thus enabling a multimodal tracking mechanism. We would therefore like to explore the addition of more sensors to the environment, such as a simple colour web-camera. This would provide valuable information about the colour of clothes worn by the principals, which would help differentiate between principals in close contact and might help rebind authentication information to a principal if the tracking is lost without requiring re-authentication.

The evaluation indicates that the scalability of the current prototype is unsatisfactory, so we wish to redevelop the tracking algorithms of the *PAISE* core component to track different people in parallel on multiple processors.

Finally, we conclude that the *PAISE* model provides a useful abstraction for authentication in smart environments, which may significantly improve the usability of a traditional authentication system. Moreover, our implementation and evaluation of the *PAISE* model indicate that it is both practical and feasible.

References

1. Cook, D., Das, S.: Smart Environments: Technology, Protocols and Applications. Wiley-Interscience (2004)
2. Focken, D., Stiefelhagen, R.: Towards vision-based 3-d people tracking in a smart room. In: Proceedings of International Conference on Multimodal Interfaces (ICMI), Pittsburgh, U.S.A. (2002) 400–405

3. Weiser, M.: The computer for the 21st century. Scientific American Special Issue on Communications, Computers, and Networks (1991)
4. Weiser, M., Brown, J.S.: Designing calm technology. PowerGrid Journal **1.01** (1996)
5. O'Gorman, L.: Comparing passwords, tokens, and biometrics for user authentication. Proceedings of the IEEE **91** (2003) 2021–2040
6. P. Srinivasan, D. Birchfield, G.Q., Kidane, A.: Design of a pressure sensitive floor for multimodal sensing. Information Visualisation (2005)
7. Roy Want, Andy Hopper, V.F., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems (TOIS) (1992)
8. Sigurjón Álrni Gudmunðsson: Robot Vision Applications using the CSEM SwissRanger Camera. Master's thesis, Institute of Informatics and Mathematical Modeling, Technical University of Denmark (2006)
9. Hansen, D., Hansen, M., Kirschmeyer, M., Larsen, R., Silvestre, D.: Cluster tracking with time-of-flight cameras. In: Proceedings of the CVPR 2008 Workshop on Time of Flight Camera based Computer Vision (TOF-CV), Anchorage, Alaska, U.S.A. (2008)
10. Kirschmeyer, M., Hansen, M.S.: Persistent authentication in smart environments. Imm-thesis-2008-16, Department of Informatics & Mathematical Modelling, Technical University of Denmark (2008)
11. Corner, M.D., Noble, B.D.: Zero-interaction authentication. In: Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MOBICOM), Atlanta, U.S.A. (2002) 1–11
12. Noble, B.D., Corner, M.D.: The case for transient authentication. In: Proceedings of the 10th ACM SIGOPS European Workshop, Saint-Emillion, France (2002) 24–29
13. Corner, M.D., Noble, B.D.: Protecting applications with transient authentication. In: Proceedings of the First ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys'03), San Francisco, U.S.A. (2003) 57–70
14. Bardram, J.E., Kjær, R.E., Pedersen, M.O.: Context-aware user authentication - supporting proximity-based login in pervasive computing. In: Proceedings of UbiComp 2003, Seatle, U.S.A. (2003) 107–123
15. Klosterman, A.J., Ganger, G.R.: Secure continuous biometric-enhanced authentication. Technical Report CMU-CS-00-134, Carnegie Mellon University (2000)