Privacy-Friendly Smart Environments

Ibrahim Armac, Andriy Panchenko, Marcel Pettau, Daniel Retkowitz
Department of Computer Science, RWTH Aachen University
Ahornstr. 55, 52074 Aachen, Germany
Email: {armac|pettau|retkowitz}@i3.informatik.rwth-aachen.de,
panchenko@cs.rwth-aachen.de

***Abstract*—In this paper we describe our approach on protecting user privacy in smart environments, particularly smart homes, which we call eHomes. These are environments with devices such as sensors, computational units, actors, which are seamlessly integrated in the environment, and objects we use in our everyday life. In order to provide more convenience to its users such environments can be personalized.**

As these environments become ubiquitous, thus supporting mobility of the users, new privacy threats arise. These are based on the digital traces and personal information which is left while visiting different environments. We provide a practical approach to minimize these traces and information disclosure by applying negotiation, identity management, and anonymous credentials. Also, we discuss the protection of eHomes from malicious users.

I. Introduction

One of the goals of our eHome project at RWTH Aachen University is to support *inter-home mobility*. This term describes the situation of users moving between multiple environments such as their home, their office, or a hotel, as shown in Figure 1. Regarding to inter-home mobility, we use the term eHome in a broader sense instead of restricting it to only households. Supporting inter-home mobility means enabling hassle-free access to these differing environments, while allowing users to keep their personal data, including preferences, for services across these environments. The preferences include the services the user wishes to use (such as heating and lighting) as well as the settings for said services (such as the preferred temperature or illumination level).

In [1] and [2] we describe our *client side personalization* approach for achieving aforesaid goal. The basic assumption here is that users carry a *mobile device* storing personal data, see Figure 1. This data is then disclosed to visited environments for personal services when needed. Inter-home mobility obviously involves the risk of privacy violation. Disclosing personal data leads users to leave digital data tracks in different smart environments, possibly in public places. In this paper, we describe how client side personalization can be supported while the privacy and security of personal data is protected.

II. eHome Background

eHomes are environments with devices such as sensors or actors connected to a hardware platform, the *residential gateway*. It runs a software platform, the *service gateway*, which allows to run *eHome services*. Below, we simply use the term *gateway* for the combination of the residential and service gateways.

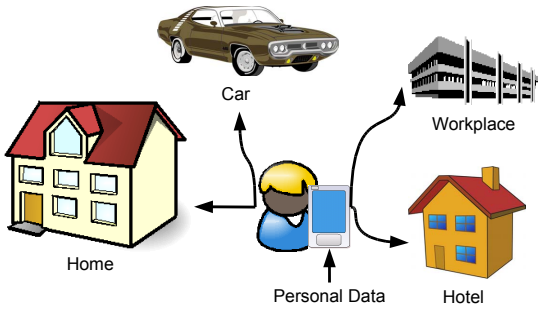

Fig. 1. Inter-home mobility

We distinguish different types of eHome services: *Basic* services act as drivers for devices, allowing us to abstract from the hardware and/or protocols used in an eHome. *Integrating* services are composed from other services, basic and/or integrating, delivering higher-order functionalities. Integrating services offering application functionalities are called *top-level* services. Application functionalities can be identified in the areas of comfort, security, infotainment, communication, health care etc. Lastly, top-level services which adapt their functionality to personal data are called *personal* services.

III. Attacker Model

We consider a local attacker with the following capabilities:

- Passively observe some portion of network traffic;
- Actively operate its own eHomes/services or compromise some fraction of honest eHomes/services;
- Actively delete, modify and generate messages.

Further we assume that the adversary cannot break cryptographic primitives. This is a standard assumption in the area of privacy protection.

Thus, our attacker model allows malicious services, eHomes, as well as colluding users.

IV. Approach and Analysis

In this section we describe our approach for protecting user privacy while enabling client side personalization of eHomes. At the end of this section we will also give some hints how we protect the security of eHomes.

A. Privacy Protection

Figure 2 shows an overview of our client side personalization approach. A user model is running on both the

[APPR09] I. Armac, A. Panchenko, M. Pettau, D. Retkowitz
Privacy-Friendly Smart Environments
In: Khalid Al-Begain (ed.): Third International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST 2009), Cardiff, UK, pp. 425-431, IEEE Computer Society, 2009
www.se-rwth.de/publications

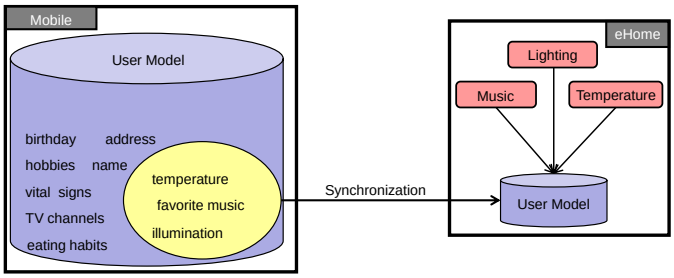

Fig. 2. Identity management and client-side personalization

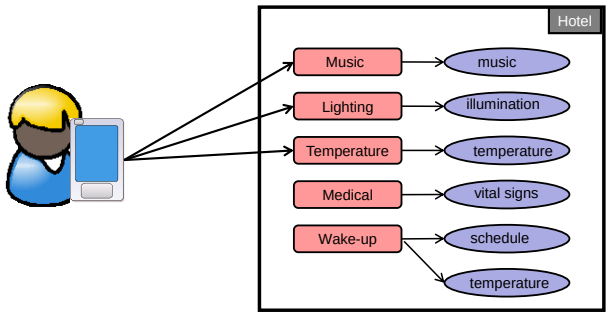

Fig. 3. Negotiation of personal services

mobile device of a user and the gateway in the eHome. The user model is responsible for managing personal data and for providing it to personal services via a unified interface. Thus, it abstracts from the kind of sensors and protocols for collecting personal data. A further task of the user model is synchronizing personal data between the mobile device and the eHome. First, when a user logs on to an eHome. Second, when the user or the services make changes on the data during a session. Third, when the user logs out.

Obviously, there exists a *contradiction* between eHome personalization and privacy protection. On the one hand, users have to provide their personal data. On the other hand, they want to hide personal data in order to protect their privacy. Based on the client side personalization approach, we developed a concept for defusing the mentioned contradiction.

Our approach is based on control of *information flow*. Basically, we enable users to minimize information disclosure combined with anonymization. Details are described below. We further assume, that users regularly (every time they need their actions to be unlinkable) change low layer identifiers of their mobile devices (e.g., MAC and IP addresses) to hinder potential attackers from identifying them. Thus, providing unlinkability involves disconnection from the network, changing of the MAC and IP address, as well as change of the location (alternatively, faked signal strengths adjustment).

1) Identity Management: In the scenario of inter-home mobility, a user will visit different types of eHomes such as his workplace, hotels or other private and public environments. Obviously, each eHome will provide varying sets of personal services to its users. This implies that a user might use different sets of personal services in different environments. For example, while a music service would make sense in a hotel, a phone forwarding service would be more appropriate at the workplace. In addition, a user might even not use the same services during each session in the same environment. In the example shown in Figure 2, the user is assumed to use all of the depicted services, namely Lighting, Music, and Temperature.

A closer look at the example reveals that each personal service requests different parts of a user's personal data. In our example, the three services need only the illumination, temperature, and music preferences for personalizing their functionalities. From this, we can conclude that only necessary parts of personal data should be disclosed to an eHome.

We realize this by introducing an *identity management* approach. We define an identity as a *subset* of the user's personal data. A user may have predefined identities such as "work", "home", "travel", or "gym" which include according preferences. We have developed a component running on a mobile device, called *Identity Manager*, which enables a user to select an identity depending on the situation and visited eHome. Only the data which is part of the selected identity will then be disclosed to the eHome, see Figure 2. Thus, we gain fine-grained user control about what data is transferred to and known by a visited eHome.

It is however not realistic to assume that a user has predefined identities for each eHome he might visit. Therefore, our approach also supports composing identities on-demand.

2) Negotiation-based Identity Management: Due to this, we have extended our approach by introducing *negotiation*. This approach enables a user to select personal services from a list provided by the eHome for his purposes. The user is also informed about what data the selected services need for personalization. Depending on his intention, he can select which services to use and, as a result, which identity to activate, i.e., which data to disclose.

An example is given in Figure 3. It shows a user who is checking in to a hotel. He is provided five personal services together with the information which personal data these services need. The user has selected three of these services, namely Music, Lighting, and Temperature. Because he does not want to disclose information about his vital signs and schedule, he decides not to use the services Medical and Wake-up.

Obviously, this kind of negotiation increases the interaction frequency. This can be inconvenient for a user if he has to negotiate the same services every time entering an environment. To reduce the interaction frequency, the user can store his last negotiation for each eHome. Next time the user enters the same eHome, the mobile device can automatically take the necessary steps enhancing the user's convenience. Of course, the user is still able to make any changes if he wishes so.

3) Service Execution on Mobile Device: Up to now, we assumed that functionalities which a user desires are realized by already running services in the visited eHome. In this case, it would be sufficient only to transfer the necessary personal data to the environment for personalization. However, there might be situations, where the visited eHome does not run the

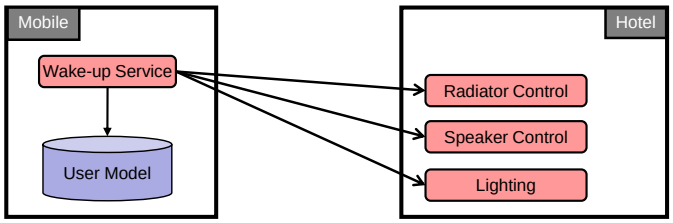

Fig. 4. Example of service execution on mobile device

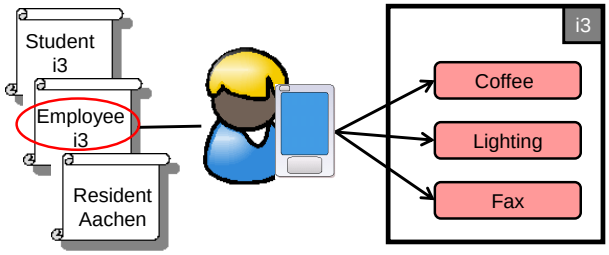

Fig. 5. Authentication with credentials

desired services. Now the question arises how a user can still be served with the desired functionalities.

We have extended our approach allowing users to take along personal services, in addition to personal data, and execute them on their mobile device when needed. Figure 4 shows an example of a business hotel guest wishing to use his personal wake-up service which realizes a comfortable wake-up procedure. As this service is not provided by the hotel, he wants to run it on his mobile device. The service however requires other functionalities which have to be provided by services on a lower abstraction level running on the gateway of the hotel. E.g., heating is required to increase the room temperature before wake-up time. Speaker control is required to play personal wake-up sound or music. Lighting is used to slowly increase the illumination level for smooth wake-up.

A distinctive feature of the wake-up service is that it can calculate the optimal wake-up time based on the user's schedule as well as on traffic and weather information. In case the wake-up service would be provided by the hotel, the user would be asked to disclose his schedule information. This can be avoided by executing the service on the user's mobile device. Thus, the personal data required by that service is not disclosed to the eHome. Instead, it is kept confidential on the mobile device[1]. Accordingly, the amount of personal data disclosed to a visited eHome can be reduced. In other words, service execution on mobile devices helps enhancing privacy protection by minimizing information disclosure. However, it implies also higher communication effort due to service interaction between mobile device and the residential gateway. Also, the energy consumption of the mobile device increases.

4) Anonymous Authentication: Up to now, we have discussed how to minimize the amount of personal data disclosed to an eHome. However, this data is usually linked to some identity. As a result, a user will leave data tracks in different eHomes, which can be linked to a single entity. Even if these data tracks correspond only to partial identities, colluding eHomes could share their knowledge for gaining more information about a user. These eHomes could either be malicious by themselves or be hacked. Moreover, also a single environment is able to recognize a user across multiple sessions and record his partial identities over time. This might be undesirable if a user appears by different identities in the same eHome over time.

[1]Note that the information transmitted to the low-level services still can be intercepted.

To overcome the aforementioned problems, we need an approach that minimizes the *traceability* and *linkability* of partial identities. By minimizing traceability we mean that a partial identity should not be linkable to a user's real identity. Of course, the disclosed data should not contain information which can reveal a user's real identity directly. By unlinkability we mean that a user's partial identity cannot be linked to any other partial identity of the same user.

Therefore, we decided to use the concepts of *anonymous credentials*, also called *minimal disclosure certificates*. In addition to minimization of traceability and linkability, anonymous credentials have also a third privacy property called *selective disclosure* [3]. Selective disclosure fits well with our goal of minimizing information disclosure, as it supports the method of disclosing only necessary data to an eHome.

We use the *idemix* anonymous credential system [4] in our prototype since it fulfills the aforesaid requirements. Basically, one or more organizations, called credential issuer or *certification authority* (CA), issue one or more credentials to a user. These credentials contain identity claims signed by the CA which can be presented to eHomes for getting access to services. A user is known to each eHome and to each CA by mutually exclusive and unlinkable *pseudonyms*. As the pseudonyms are unlinkable, also the corresponding partial identities are unlinkable. A further property of idemix is that the pseudonyms are not traceable. That means that an eHome cannot find out the real identity of a user showing a credential based on this credential. Even if the same credential is presented to an eHome multiple times, the eHome cannot recognize these repeated presentations.

Consider the university example in Figure 5. The user has got three different credentials issued by different CAs. The bottom credential issued by the city council of Aachen proves his residence in Aachen. The other two issued by the Department of Computer Science 3 at RWTH Aachen University (i3) prove his student respectively employee status. Depending on his intended activities, the user selects one credential for authentication. In the example, he selects the employee credential as a student is not allowed to use the Fax service. The eHome grants him access to the three services after verifying that the user has shown a credential attesting his employee status. Consider that the eHome only knows that an employee has shown the credential. It does, however, not know *which* employee. It would be also possible to use

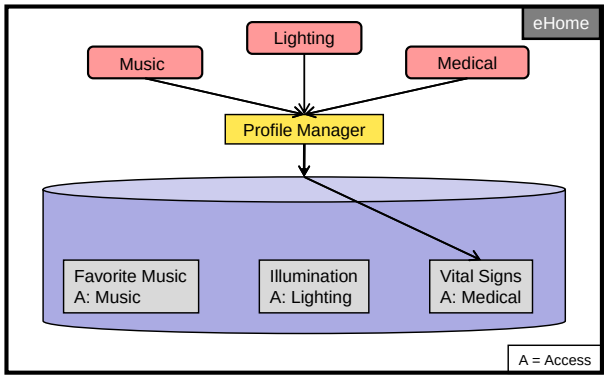

Fig. 6. Confidentiality by selective access

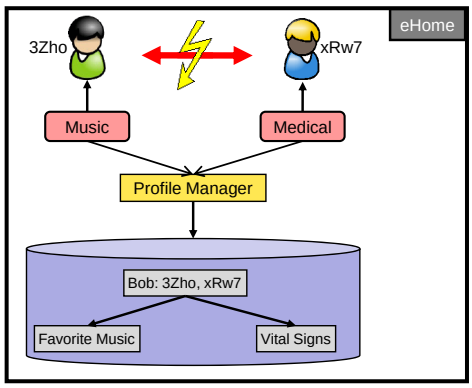

Fig. 7. Unlinkability between services

different credentials for accessing different services. However, since we further make use of session credentials (their need will be explained later), at the moment there is a need to select credential(s) that allow access to all the required services.

Anonymous credentials do not only support the protection of privacy. They can also be used to realize access control, as will be described in Section IV-B.

5) Confidentiality by Selective Access: Minimizing the amount of personal data disclosed to an eHome in an unlinkable and untraceable way helps us protecting privacy. However, we should further have a look at the data which is nevertheless disclosed to an eHome and managed there by the user model.

For the following discussion we assume that we trust services less than we trust the user model in an eHome. This assumption is based on the idea that the user model can be delivered by a trusted organization while services can be obtained from various vendors. It is a matter of common knowledge that various companies are interested to collect personal information. This information is then used for different purposes, such as spam and retail.

As a result, we are interested in minimizing the amount of personal data provided to services. This can be done by granting a service access to only those preferences of a user which are absolutely necessary for its own functionalities. Other preferences of the same user should not be accessible for that service. For achieving this goal we have developed an access control mechanism which allows a user to interactively grant access to specific preferences in his profile. For each attribute, he can specify which services are allowed to access it. All other services are not permitted to access the data. This technique is called *confidentiality by selective access* [5].

We have developed the *Profile Manager* which realizes the confidentiality by selective access approach (see Figure 6). For every preference attribute in the user model an *access control list* indicates which services are granted access to this attribute. If for example the Medical service would request the values of the vital signs, the Profile Manager would grant access to this attribute because the Medical service is included in the access control list, indicated by "A: Medical". If other services such as Lighting or Music would request for the vital signs, the Profile Manager would reject their request respectively.

6) Unlinkability Regarding Services: Confidentiality by selective access avoids services to get more information about a user than necessary. However, multiple services could still collude and share their data. To avoid this, we have extended the Profile Manager. It generates random and *mutually distinct pseudonyms* for every user/service pair and passes them to the corresponding services. In the example shown in Figure 7, the user with the *main pseudonym* "Bob" is using the services Music and Medical. He is known to these services by the pseudonyms "3Zho" and "xRw7" respectively. So, the services do not know that they are used by the same user. In addition, the services cannot discover what other services are being used by the same person.

Combined with confidentiality by selective access, this greatly reduces linkability. Since the services do not know the main pseudonym they are assigned to, they would have to compare the preferences to see if they are assigned to the same person. If the user has smartly chosen the access permissions, allowing access to an attribute only if a service absolutely needs it, then the overlap of attributes shared by the services will be very small to nonexistent. Of course, this approach works well only if the preferences provided to single services do not identify individuals directly.

B. Protection of eHomes

In the last section we discussed how eHomes can be personalized while the privacy of mobile eHome users can be protected. It is however also important to protect eHomes themselves. We will discuss in the following two kinds of access control for services running in an eHome.

The first kind of protection is needed for services being accessed by *other services* due to our layered service architecture. Consider Figure 8 which depicts the example of the Personal Lighting service. It can personalize the illumination level based on user preferences. Therefore, it uses the Illumination service which provides illumination based on artificial or natural lighting. In case there is bright sunlight outside, the roller blinds can be used to control the illumination level. In other cases, especially during the night of course, artificial lighting based on lamps is used for this purpose.

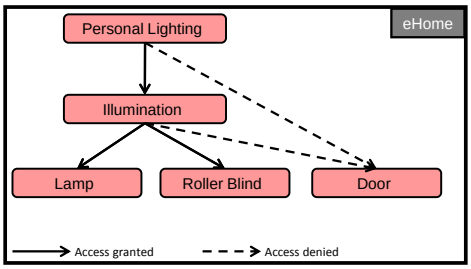

Fig. 8. Access control between services

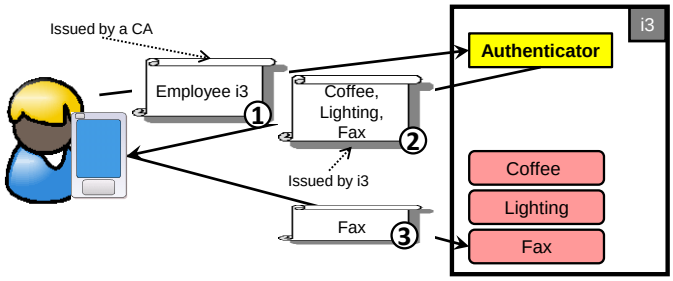

Fig. 9. Access control using session credentials

The layered service architecture enables services to use other services for realizing their functionalities. The continuous lines in Figure 8 depict that the corresponding service/service interactions are necessary and will be enabled by the system. However, considering the Personal Lighting or Illumination services being malicious, we need an approach which denies access to other services which might be critical for an eHome's security, such as the service for opening doors. Therefore, we use service roles for restricting a (malicious) service's ability to misuse low-level services.

Our approach is based a the *configuration graph* which holds the current composition information for each service in an eHome, for more details on the configuration graph we refer to [6]. Assuming that the eHome administrator trusts the specifications of the services he installs in his eHome, a service gets bindings only to those services which are necessary for its functionality. However, if a service gets malicious after deployment (e.g., by being hacked), it could try to get access also to further services. To avoid this, we generate roles for each service which allow it to gain access only to those services to which it is bound in the configuration graph. Considering the composition in Figure 8, the Illumination service will be assigned to a role which permits access only to the Lamp and Roller Blind services. Thus, it cannot use other services such as the Door service.

The second kind of protection is needed for services being accessed by *users*. The eHome owner/administrator might not want to allow each person to use each service. On the first stage of our work, we used *role-based access control* (RBAC) for this purpose. In a home environment, users could be assigned to roles such as "adult", "child", or "guest". In a university building, there could be roles such as "professor", "employee", or "student". Every role was bound to a set of eHome services. A user assigned to a specific role gained permission to use only those services which were bound to the role. E.g., in the university scenario employees are allowed to use the Fax service while students are not.

However, on the one hand RBAC requires a user to be known to an environment in advance. On the other hand, it does not provide privacy protection. To overcome these problems, we decided to realize access control for user-service interactions based on idemix. Our approach works as follows. An eHome can require a user to prove certain properties before granting him access to services. A precondition here is that a certification authority has issued the user idemix credentials attesting him these properties in advance. Based on the credentials shown, the eHome can decide to grant access to the negotiated services, see Section IV-A2.

Figure 9 depicts an example of a user wishing to use the Fax, Coffee, and Lighting services at the university department (i3). Therefore, he presents his credential attesting him to be an employee of i3. In fact, the user does not present the credential but proves based on a *zero-knowledge protocol* that he is in possession of an according credential (for more details on idemix protocols see [4]). As this information is sufficient for granting access to the required services, i3 issues him a new credential, called *session credential*. The session credential grants the user access to only those services which have been negotiated with the environment. This credential must then be used for interacting with the services. In the depicted example, the user can prove that he is authorized to use the Fax service by presenting his session credential.

Please note, that it is also possible to use high-level credentials (as, e.g., "Employee i3") at this stage. However, there are situations where session credentials are necessary or they simply offer more convenience. For example, they can be used to set a multitude of attributes allowed within a session (e.g., rooms that can be entered by a hotel guest during his stay). Additionally, the n-times spendable e-tokens [7] can be used for privacy-friendly accounting within a hotel (e.g., e-token for visiting sauna n times). This way the hotel will only know that the guest could have at most n times visited the sauna, without knowing how many times exactly and at what time. Consequently, it would not be known who else was in the sauna at the same time.

The use of session credentials brings some advantages compared to RBAC. First, the session credential can be generated on-demand according to the negotiated services, also for users not being known to the eHome in advance. Second, the session credential can be different for each session of the same user. Third, as the user proves for every service its access rights separately, it is difficult for the system to match him to a group (if the service is accessible by more than one group).

V. Implementation

For realizing access control on services, we have implemented an *interceptor*-approach with AspectJ, an aspect-oriented extension to Java, see Figure 10. It intercepts re-

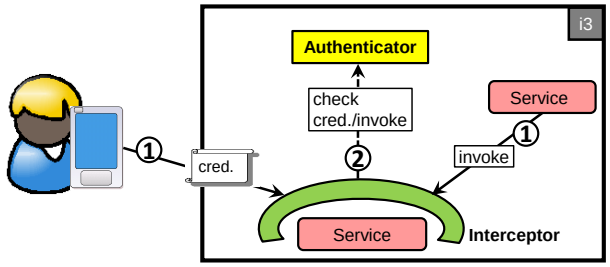

Fig. 10. Access control realized by an interceptor

quests from both users and other services (1) and asks the Authenticator for permitting or refusing the request (2). The Authenticator checks either the user proof based on his session credential or the invoking service's role. Depending on the Authenticator's response, the request is permitted or denied. The advantage of an interceptor is that services' implementations do not need to care about access control. Furthermore, if changes need to be made to the access control policies, they can be applied to the interceptor being valid for all services.

The implementation of our overall concept is based on Java on top of the Eclipse Embedded Rich Client Platform, which uses OSGi for providing a plug-in based framework for eHome services. As a mobile device we use a PDA, Dell Axim 51v, with WLAN running the IBM J9 Java VM, the only free available VM with satisfactory properties for OSGi development on Windows Mobile 5. The communication on top of WLAN is based on JXTA, a language-independent P2P protocol. Furthermore, we implemented our own RMI-like communication over JXTA, called "SimpleRMI". Our eHome prototype includes the eHomeSimulator [8], a 2D environment which enables to simulate smart environments with different kinds of sensors and devices as well as user behavior. We have tested our results with several personal services such as Lighting, Music, TV, Temperature, and Wake-up. A tool called *eHomeAdministrator* has been developed for managing access control and RBAC settings [2].

Of course, privacy protection requires also encryption of personal data both when transferring it over WLAN and storing it on the residential gateway. We have used the *Bouncy Castle Crypto APIs* [9] for implementing a hybrid encryption mechanism for data transmission over JXTA. Moreover, personal data in the eHomes is also stored encrypted.

An important aspect of our approach is its performance which we have evaluated with two Java VMs running on a laptop (AMD Turion 64 X2 Mobile TL-60, 2x2.0 GHz), namely Sun's VM and IBM's J9. The results, together with 95% confidence intervals, are shown in Figure 11 and 12. The first one shows the time needed for the issuer (e.g., an eHome in case of a session credential) and the receiver (e.g., a user) on both VMs respectively. Figure 12 depicts the performance results of both VMs when showing a credential. Again, it is shown how long the client (e.g., a user as prover) and the server (e.g., a eHome as verifier) need for their computations. Also the total duration of proof computation and verification

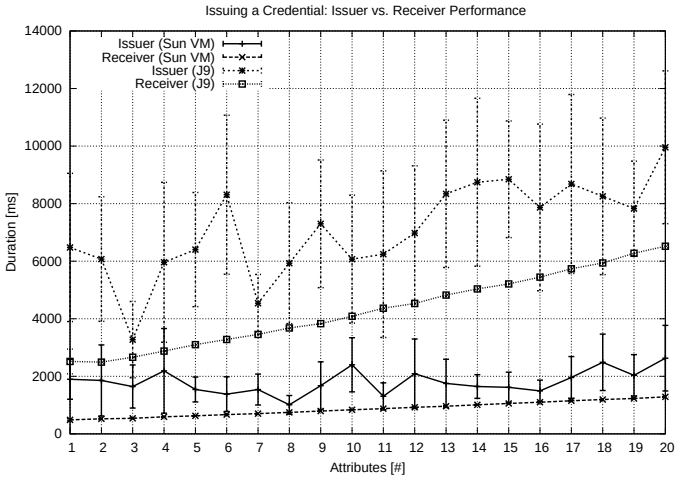

Fig. 11. Idemix performance: issuing a credential

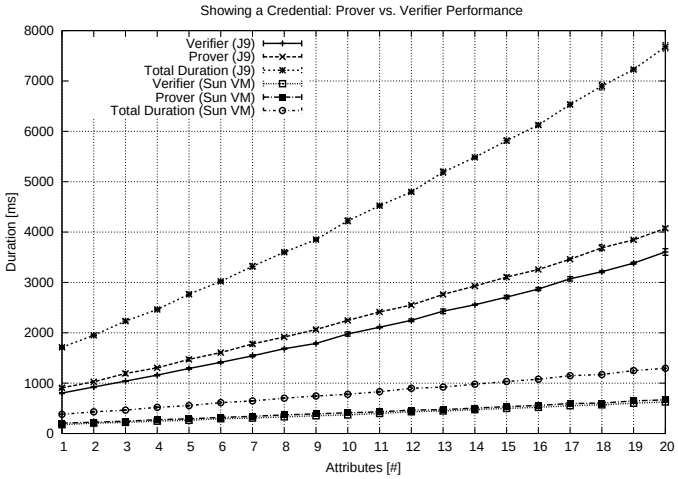

Fig. 12. Idemix performance: showing a credential

is depicted. Both evaluations have been done with credentials containing up to 20 attributes.

Several conclusions can be derived from the evaluation. First, the number of attributes affects the performance of idemix. Second, J9 is significantly slower than Sun's VM. Unfortunately, we could not find a Sun VM for mobile devices which has similar capabilities as J9. However, the comparison shows that there is great potential in enhancing the performance on the VM level. Third, even the tests with Sun's VM shows that the use of idemix is not as efficient as desired. Since a session credential is issued to user only once at the beginning, the lack of performance here is tolerable. However, the session credential is used for anonymous service access during a session. Even though each time only one attribute is shown, namely the name of the service to be accessed, and the total duration of ca. 0.4 sec. for proving an attribute is tolerable (see Figure 12), the performance with J9, especially on a PDA, is much more slower.

It is however promising that idemix has potential for optimization as a member of the idemix group at IBM Zürich

confirmed. The performance of their reference implementation is not optimized yet and future versions are expected to run efficiently on mobile devices such as PDAs. Furthermore, we can expect future PDAs to be equipped with enhanced computational properties.

Furthermore, our approach enables users to select between anonymous and non-anonymous authentication as anonymity may not be necessary in some eHomes such as the own household.

VI. Related Work

Langheinrich introduces *pawS* [10], a privacy aware system for ubiquitous computing environments. The focus of pawS lies on providing users a policy-based "privacy-enabler" which discloses data to ubiquitous services only if these services agree to use personal data according to the user's preferences. This approach however depends on trusting the services for following their policies.

Gaia is a project in the domain of smart environments. Their security and privacy approach envisions the use of roles, policies and credentials (generic, restricted, and non delegable) [11]. Several kinds of policy-based access controls have been realized (role-based, discretionary, and mandatory). Access control between components is done by credentials. The credentials can contain activated roles of a user, however, they are not anonymous. Thus, they do not provide privacy to their users. Additionally, the approach makes use of an anonymization protocol in order to hide the location of the user. This protocol relies on redundant routing (similarly to the privacy approach for MUSDAC [12]) through non collaborating peers, which is not possible in our scenario.

Schäfer et al. introduce an architecture for a secure profile management middleware with several components: security manager, profile manager, device manager and several authenticators [13]. The architecture is based on the OSGi framework. The main principle they realize is a ticket service for secure profile and user/application access right management. However, the approach merely concentrates on the access control and does not provide any protection for the privacy by ensuring unlinkability and anonymity of its users.

Other approaches are less directly applicable in our scenario. Most of them present only a proposal without any practical implementation (e.g., [14], [15], [16]) or a kind of survey (e.g., [17]). Some of them make use of multi-agent systems [18] or combine it with an anonymizing proxy [16].

VII. Conclusion and Outlook

This paper provides a practical approach for protecting user privacy in eHomes. The emphasis is on the user mobility, since with the ubiquity of smart environments the privacy threats are especially high. We achieve privacy protection for the eHomes users by minimizing disclosure of private information, omitting the disclosure at all when possible, and providing unlinkability between single actions. We practically tested our approach by a proof-of-concept implementation and provided its evaluation. We inferred that our approach provides potential for performance improvement. Finally, it should be kept in mind that the weakest part of the whole protection chain is the user: non cautious behavior and explicit information dissemination cannot be compensated by any technical means.

References

[1] I. Armac and D. Evers, "Client Side Personalization of Smart Environments," in *SAM 2008: Proc. of the 1^{st} Intl. Workshop on Software Architectures and Mobility at ICSE 2008*. ACM, 2008, pp. 57–59.

[2] I. Armac and D. Rose, "Privacy-Friendly User Modelling for Smart Environments," in *Proceedings of the The Fifth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2008)* . ACM, 2008.

[3] S. Brands, "A Technical Overview of Digital Credentials," Credentica, Tech. Rep., 2002.

[4] J. Camenisch and E. V. Herreweghen, "Design and Implementation of the idemix Anonymous Credential System," in *CCS '02: Proceedings of the 9^{th} ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 21–30.

[5] J. Schreck, *Security and Privacy in User Modeling*. Kluwer Academic Publishers, 2001.

[6] D. Retkowitz and M. Stegelmann, "Dynamic Adaptability for Smart Environments," in *Distributed Applications and Interoperable Systems, 8^{th} IFIP WG 6.1 International Conference (DAIS 2008)*, ser. LNCS, R. Meier and S. Terzis, Eds., vol. 5053. Springer, 2008, pp. 154–167.

[7] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to Win the Clone Wars: Efficient Periodic n-Times Anonymous Authentication," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006)*, Alexandria, Virginia, USA, 30 Oct - 3 Nov 2006.

[8] I. Armac and D. Retkowitz, "Simulation of Smart Environments," in *Proceedings of the IEEE Intl. Conf. on Pervasive Services 2007 (ICPS'07)*. IEEE Press, 2007, pp. 257–266.

[9] Legion of the Bouncy Castle, "Bouncy Castle Crypto APIs," http://www.bouncycastle.org/java.html, 2009.

[10] M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments," in *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, ser. LNCS, vol. 2498. Springer-Verlag, 2002, pp. 237–245.

[11] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane, and M. D. Mickunas, "Towards Security and Privacy for Pervasive Computing," in *Software Security - Theories and Systems*, ser. LNCS, vol. 2609, 2003, pp. 1–15.

[12] R. S. Cardoso, P.-G. Raverdy, and V. Issarny, "A Privacy-Aware Service Discovery Middleware for Pervasive Environments," in *Trust Management*, ser. IFIP International Federation for Information Processing, vol. 238, 2007, pp. 59–74.

[13] A. Marin, W. Mueller, R. Schaefer, F. Almenarez, D. Diaz, and M. Ziegler, "Middleware for Secure Home Access and Control," in *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE Computer Society, 2007, pp. 489–494.

[14] R. S. Cardoso and V. Issarny, "Architecting Pervasive Computing Systems for Privacy: A Survey," in *WICSA '07: Proceedings of the Sixth Working IEEE/IFIP Conference on Software Architecture*. Washington, DC, USA: IEEE Computer Society, 2007, p. 26.

[15] a. P. Sousa, Jo "Challenges and Architectural Approaches for Authenticating Mobile Users," in *SAM '08: Proceedings of the 1st International Workshop on Software Architectures and Mobility*. ACM, 2008, pp. 15–20.

[16] S. O. Hwang and K. S. Yoon, "Privacy Protection in Ubiquitous Computing Based on Privacy Label and Information Flow," in *Computational Science and Its Applications - (ICCSA 2004)*, ser. LNCS, vol. 3044. Springer, 2004, pp. 46–54.

[17] A. Görlach, A. Heinemann, and W. W. Terpstra, "Survey on Location Privacy in Pervasive Computing," in *Privacy, Security and Trust within the Context of Pervasive Computing*, ser. The International Series in Engineering and Computer Science. Springer, 2004, pp. 23–34.

[18] A. J. Blazic, K. Dolinar, and J. Porekar, "Enabling privacy in pervasive computing using fusion of privacy negotiation, identity management and trust management techniques," in *ICDS '07: Proceedings of the First International Conference on the Digital Society*. IEEE Computer Society, 2007, p. 30.