Trustworthy User-Centric Identity Management Based on Personalised Java Cards

Mario Hoffmann, Jan Peter Stotz

Fraunhofer Institute for Secure Information Technology,
Darmstadt, Germany

{hoffmann;stotz}@sit.fraunhofer.de, http://www.sit.fraunhofer.de/

***Abstract*—Personalised ambient-aware services are on the roadmap of most research programmes. In order to take particularly advantage of mobile and wireless communications, security and privacy enhancing technologies need to be technically introduced in both the design and the implementation phase at the very beginning.**

Here, user-centric identity management becomes a key technology in order to control access to information such as the user's current location, his or her personal environment, and their personal profiles. In our approach especially access to profiles containing virtual identities, personal preferences, and service histories are protected by a Java Card enabled authorisation protocol and ticketing system.

This ticketing system is embedded in a framework adding user-centric identity management to a context-aware mobile services platform. This platform has been already designed and partially implemented to provide and aggregate location based services. Both, the ticketing system and the appropriate parts of the platform will be described in detail in this paper.

Key words: User-centric identity management, privacy enhancing technologies, ambient awareness, mobile services, service oriented architectures

1. INTRODUCTION

The ambient world is divided into two major approaches. The first approach supports users receiving individual context-aware information by the surrounding environment. The second one enables the surrounding environment to get personal information about the users. Both approaches are part of the same medal in order to make the ambient experience as individual and context-related as possible. From a privacy and security point of view, however, this vision leads inevitably to the question: Who or what has when and why access to the user's personal profiles, preferences, and service histories?

Here, let us assume that end-users in the future will still have several different mobile devices with particular restrictions concerning encryption capabilities, integrated security mechanisms, and trustworthyness. Let us further predict that (virtual) identities of (mobile) users will be the more attractive to attackers the more personal and context-aware information is stored (identity theft) – as recent serious attacks against companies storing credit card information in the US already show. Then we can derive the following principal requirements:

- security mechanisms have not only to be introduced at service provider and network operator level but also at end-user level (multilateral security),
- privacy and data protection has to be inherently supported the more the ambient experience becomes reality in order to reasonably balance control of individual profiles,
- mobile users have to be transparently supported by privacy enhancing technologies.

Smartcards such as Java Cards is such a technology. In our approach described in detail in chapter 4 Java Cards first of all support the last requirement by securely storing particularly sensitive information such as private keys. The second requirement is addressed by providing tickets to authorise temporary access to personal information stored at dedicated data bases. Taking advantage of Public Key Infrastructures (PKI) by all parties involved even ensures the first requirement by enabling mutual trustworthy authentication particularly between service providers and end-users.

Thus, we have already identified the three basic steps towards user-centric identity management. As the general introduction of user-centric identity management to open mobile environments has been already addressed in [3] this paper focusses on the implementation and integration using personalised Java Cards.

2. STATE OF THE ART

Taking into account the 5th and the current 6th Framework Programme (FP5, FP6) the European Union has founded several projects dedicated to context-awareness and enhanced security and privacy technologies. RAPID ([4]) and PAMPAS ([5]) in FP5 for example have proposed roadmaps for advanced research in privacy and ID-Management respectively mobile privacy and security. Based on the results PRIME (cf. [6]) has been

started in 2004 in order to fulfil the roadmaps by empowering individuals to control their private sphere and manage their identities. Related projects that have also started during FP6 are the WWI, SEININT, UBISEC and SWAMI (cf. [10], [7], [9], and [8]). Strategic objectives addressed by these projects are:

- further development of the wireless world B3G,
- trusted and dependable security frameworks,
- advanced infrastructures for secure large-scale user mobility based on Smart Card technologies
- and so-called dark scenarios possibly implied by ambient intelligence (AmI).

Research on ambient-awareness and AmI will be even more intensive when EU's 7th Framework Programme starts in 2006.

Besides the most relevant European projects one of the best known projects in the US is *my*Campus, a development at Carnegie Mellon University (CMU), PA (cf. [2]). Here, *my*Campus is a Semantic Web environment for context-aware mobile services aimed at enhancing everyday campus life at CMU.

Partially inspired by these research activities the R&D group Secure mobile Systems at Fraunhofer SIT has found its niche by developing *MOBILE – Secure Services for Mobile Users* since June 2001. As a platform for context-aware mobile services MOBILE features multimodal access, multi-agent technology and last but not least a security framework for wireless and mobile services ensuring privacy.

The next chapter describes derived design principles before chapter 4 introduces Java Cards to MOBILE to particularly support the user's identity management. (For your interest: An enhanced demonstrator has been shown at CeBIT fair 2005 and MobiCom2005[1].)

3. DESIGN PRINCIPLES

The reference model of the MOBILE platform is devided into two major domains: the *trust domain* where access to personalised information is comprehensively supervised by a personal assistant (also: personal agent, cf. [1]) and the *service domain* expected not to be controlled by the end-user. The latter one consists of three areas shortly sketched:

- provider-centric enabling services such as directory, ontology, and PKI services,
- value-added services such as particular mobile enterprise, streaming, and application services,
- and basic services such as localisation, m-commerce, mobile gaming, and emergency services.

The former one, the trust domain, discussed in this paper and shown in the figure below comprises basically two clusters:

- the user domain simply comprising personalised Java Cards, the mobile device, and a client proxy sending the task requests to the personal assistent
- and user-centric enabling services being part of the *Personal Assistant*.

For a better understanding: Task requests are one of the key concepts realised by MOBILE in the future in order to basically save time and money. Tasks inherently comprise several (user defined) subtasks that can be handled by the personal assistant automatically when the user is offline for example. A typical task is travelling: Let us imagine one has entered a conference in Aalborg in September in his business calendar. Then typical subtasks are finding hotels considering budget constraints, offering reasonable alternatives for travelling (i.e. by train, by plane, by car), providing a city map, and identifying points of interest with respect to personal preferences.

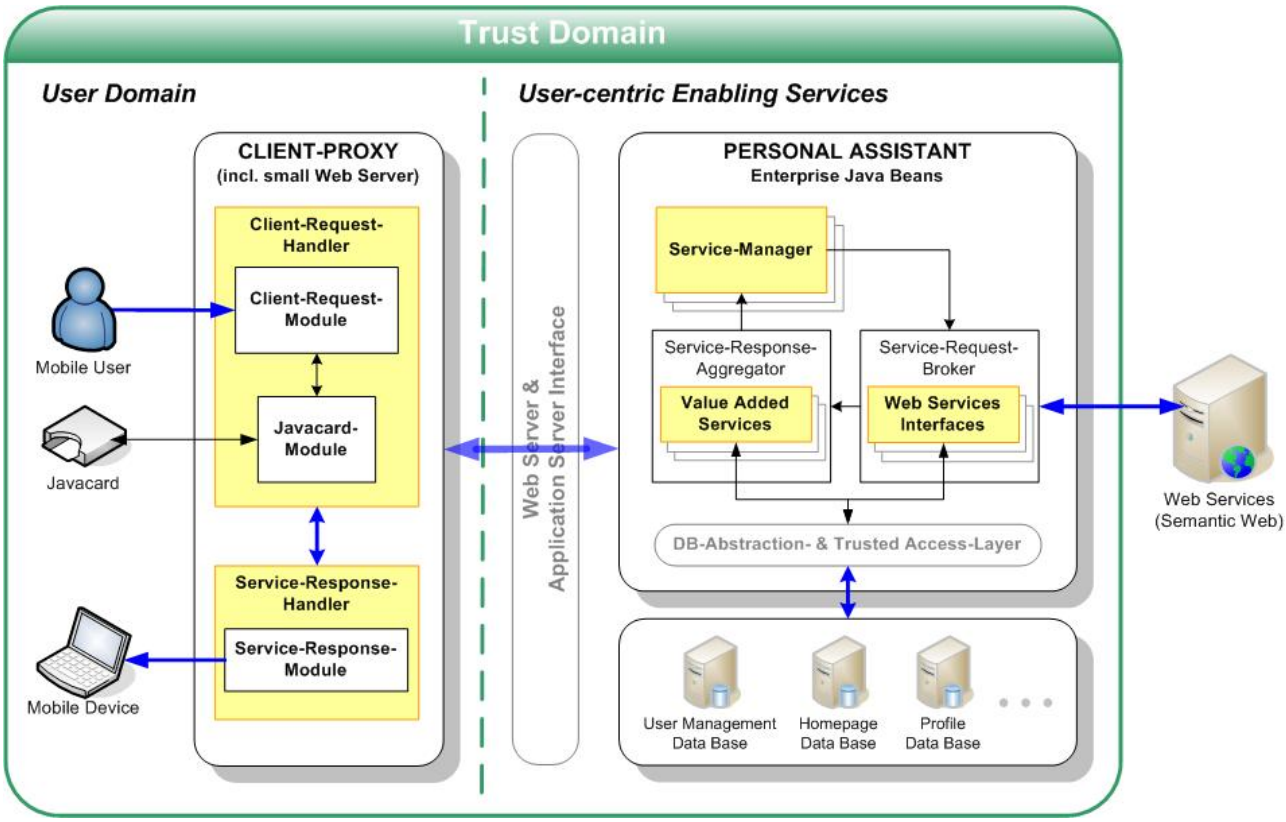
All subtasks determined by the *Service Manager* are operated by the *Service Request Broker* and gathered by the *Service Response Aggregator* automatically. During this process the subtasks several times need access to the user's personal profiles, his preferences, and service histories. As the corresponding data bases are encrypted and access is protected by a trusted access layer the user is asked for authorisation. Not to bother the user with continuous requests for every single access and to enable offline activities a task-based ticketing system has been introduced (cf. next chapter). As soon as the user is online again or is checking the current progress of the task (temporary) results will be presented device dependent. Note: The final decision of all eCommerce related (sub)tasks is – at this stage of the project – always up to the user.

The following chapter discusses in detail now different opportunities how users can take advantage of using Java Cards in the MOBILE platform considering both online and offline implications.

4. THE CORE SECURITY SYSTEM

Smartcards are an easy and reasonable way for securely storing small amounts of sensitive data – for example private keys. However, smartcards of the latest generation are even under-worked by these tasks. Open card smartcard systems like Suns Java Card allow the installation of self written applications. These so called applets combine the aspects of secure storage and secure processing of sensitive data. Even if multiple applets are installed on the same card every applet is sealed off in its own partition by default.

[1] MobiCom2005, Cologne, Germany, Aug 29th-Sep 2nd, http://www.sigmobile.org/mobicom/2005/



4.1. Identity Management using Smartcards

There are several possibilities of using Java Cards in an identity management system. The direct approach, storing sensitive data completely on card, can be understood intuitively. However, the flexibility of such a system is restricted to certain extent. A lost card signifies a complete loss of all stored data on that card. This problem could be solved by introducing backup cards, but that would require regular recurrent and sustainable synchronisations between all cards by the user. For these reasons it makes more sense using a client server system that allows centralised data storage and automatic synchronisation when connecting to the server. Of course, this server side storage of sensitive user data has to be in a secure way from the user's point of view. If we assume that this database stores the user's data reliably, there is no need to keep copies on the user's Java Cards and it allows us to exceed the limitations set by the available persistent memory on card.

There are several requirements identified in the next section that have to be met by this client server system.

4.2. Requirements

Even though our identity management system assigns one account to one person there is great demand for having multiple smartcards per user. Having one card for every device allows the user to install the smartcard into the device instead of constantly swapping one card between multiple devices. As another advantage, smartcards embedded inside of a device, are lost much less frequently than small (may be SIM sized) smartcards. These requirements lead us to the conclusion that our identity management system has to include a card management system that allows the user himself to add new smartcards to the system or block lost ones.

4.3. Multiple cards – multiple problems

The easiest approach allowing multiple cards would be to create smartcards with identical content and a unique card ID on it. This ID can be stored in a database within the identity management server together with the card state (working or blocked). But this approach has an important problem: It breaks with our target in creating a user-centric system. There is no way to assure that smartcards blocked once will remain blocked.

The system security is depending on the security of the user and card management database (which is in general not assumed to be under the user's control). One way to solve this problem is the usage of public key cryptography.

4.4. The resulting system

On adding a new Java Card to the system it generates a new RSA key pair on card. The card's public key is stored in the card management database. For securely identifying the public keys in the database as authorised by the user, each public key is stored together with a

keyed-hash message authentication code (HMAC) of the card's public key, the card ID and a secret pass phrase. This pass phrase can be chosen by the user when registering the first card into the system. Each card stores a copy of the pass phrase and uses it for on card verification of other card's public keys.

Because of a relatively small number of active Java Cards belonging to a user, every card can store a copy of all the public keys belonging to the user. This on card copy can be synchronised whenever there is a "connection" between the Java Card and the server.

4.5. Key management and data encryption

On this basis it is not hard to set up the encrypted data storage in the style of an encrypted file system like Microsoft's Encrypting File System (EFS):

Before data is stored in the profile database it gets encrypted with the data encryption key – a symmetric, random key. For maximum security, key generation and data encryption should be done by the Java Card itself. The data encryption key then is stored in the database as many times as the current user has active cards. Each copy is encrypted with one of the card's public keys.
Reading data from the profile database can be achieved in four steps:

1. Determine the data encryption key.
2. Find the stored data encryption key which is encrypted with the public key of the currently used Java Card.
3. Load the encrypted data encryption key onto the Java Card and decrypt it.
4. Decrypt the profile data using the Java Card.

4.6. The server side assistant

This so far described system relocates all decryption and encryption activities to the Java Card on the client system; but now the server side assistant comes into play. Every user has its own assistant-instance running on the server, managing several different tasks given by the user. Some tasks can be executed immediately after committed to the assistant, others may be timed for execution on occurrence of an event or a specified time and date. On execution time of these delayed tasks, we can not presume that one of the user's Java Cards is connected to the server right in that moment the execution takes place (and the data would be needed). Therefore, the assistant has to be able executing tasks without an active connection to a client and it's Java Card. For that reason it has to be able to decrypt profile data as a user-proxy. This is implemented by using a *ticketing system.*

4.7. A ticket, please

Tickets are containers for data encryption keys, encrypted with the public key of the database's trusted access layer – an independent system that acts in between the assistant and the database. The advantage is that tickets may include usage constraints and a maximum lifetime.

Instead of decrypting the data by the Java Card, now only tickets are created by the Java Card and sent to the assistant which stores them until they are needed.

4.8. Database trusted access layer

If a task handled by the assistant requires data from the database and there is a suitable ticket present, both, request and ticket are sent to the database trusted access layer. There, the ticket is being decrypted by using the database's private key. This unveils the secret data encryption keys which are now used for decryption of the acquired user data from the user's profile database.

The database trusted access layer is implemented as a stateless system. This means that all secret data, i.e. the content of the ticket with its data encryption keys and the unencrypted user profile data are deleted from memory before the next request from any assistant can be worked off.

5. CONCLUSION

As discussed there are different opportunities to take advantage of using Java Cards for trustworthy user-centric Identity Management. We decided to go for a hybrid approach as we needed (1) a dedicated, generic, and user-controlled security token to store most sensitive data, (2) cryptographic capabilities to encrypt and decrypt personal profile data when the user is online, and (3) the issue of temporary authorisation tickets for off-line phases. Moreover, the possbility for the user having exchangeable dedicated Java Cards for each mobile device assures flexibility and usability.

Nonetheless further tests and integration into different mobile devices has to be concerned as the current system focuses on Laptop PCs which most easily can be enhanced by Smartcard readers. Organisers (PDAs) can also be enhanced by Smartcard Readers by so-called Jacket solutions. Only Smartphones or cell phones – as the most important device class – rely on built-in Smartcards. However, several SIM cards are already Java enabled so that a certain sustainability is ensured once Mobile Network Operators identify trustworthy context-aware mobile services as a promising business case.

REFERENCES

[1] Bemmel, G. van, Hoffmann, M., Teunissen, H., *Privacy and 4G Services: Who do you trust?*, Conference of the Wireless World Research Forum, New York, 2003

[2] Gandon, F. L., Sadeh, N.M., *Semantic Web Technologies to Reconcile Privacy and Context Awareness*, Computer Science Technical Report CMU-CS-03-211, Carnegie Mellon University, Pittsburg, PA

[3] Hoffmann, M., *User-centric Identity Management in Open Mobile Environments*, Privacy, Security and Trust within the Context of Pervasive Computing, Robinson, Ph., Vogt, H., Wagealla, W. (Eds.), pp 99-104, Springer, 2004, ISBN 0-387-23461-6

[4] Huizenga, J. (Ed.), *RAPID – Roadmap for Advanced Research in Privacy and Identity Management*, IST-2001-28210, Final Report, http://www.ra-pid.org

[5] Hulsebosch (Ed.), *PAMPAS – Pioneering Advanced Mobile Privacy and Security*, IST-2001-37763, Final Roadmap, http://www.pampas.eu.org

[6] *PRIME – Privacy and Identity Management for Europe*, EU 6th Framework Program, Project Identifier IST-2002-507591, *Privacy and Identity Management for Europe*, http://www.prime-project.eu.org/

[7] *SEINIT – Security Expert Initiative*, EU 6th Framework Program, http://www.seinit.org/

[8] *SWAMI - Safeguards in a World of Ambient Intelligence*, EU 6th Framework Program, http://swami.jrc.es/pages/index.htm

[9] *UBISEC - Ubiquitous Networks with a Secure Provision of Services, Access, and Content Delivery*, EU 6th Framework Program, http://www.c-lab.de/ubisec/

[10] *WWI – Wireless World Initiative*, EU 6th Framework Program, http://www.wireless-world-initiative.org/