Safeguarding Personal Data with DRM in Pervasive Computing

Adolf Hohl[1] and Alf Zugenmaier[2]

[1] Institute of Computer Science and Social Studies
University of Freiburg, 79098 Freiburg, Germany,
adolf.hohl@iig.uni-freiburg.de,
[2] Microsoft Research Cambridge
alfz@microsoft.com

Abstract. Pervasive computing can be divided into computing on personal wearable devices and computing in a smart infrastructure. When a wearable device communicates personal data for further processing to the infrastructure, privacy concerns arise. This paper presents an approach to dispel concerns relating to improper use of personal data based on digital rights management technology. A prototype implementation of this approach in a smart hospital environment is described.

1 Introduction

The paradigm of ubiquitous and pervasive computing [1] leads to a much greater intrusion of information and communication technology into the personal life of everyone than what we experience today. The users of pervasive computing will use many smart personal objects, in addition, many services will be provided by a smart environment that will surround us. However, fears of users about the misuse of their personal data prevents the acceptance of these services and technologies. This is especially the case, when an agent running on a personal digital assistant is acting on behalf of the user and can autonomously release sensitive information to communicating partners such as service providing devices in the environment. Nearly everybody has had experience of misused personal information in the Internet such as unwanted advertisements and spam. This is only the tip of the iceberg. More serious abuse of the information may involve selling it to rating agencies, resulting in unwanted "personalization" of prices, interest rates, denial of credit, etc.

Therefore, it is essential that devices providing services handle their users' personal data with care. If it is not possible to ensure this, fear of misuse and privacy concerns remain with the user.

1.1 Problem Statement

In this paper, we address the problem of giving users of pervasive computing environments more control over their data after they are transmitted, e.g., during

the use of a service or an application. Privacy issues can never be addressed completely without looking at the application domain [2]. Therefore, we make use of the scenario provided by the project EMIKA at the University Hospital of Freiburg [3][3]. In the hospital scenario, patients are equipped with a smartcard which can store the patients' health history or parts thereof[4]. In this scenario, patients can have access to the content of their smartcards and supplemental information, which is linked to other sources of information on this card or external to it. EMIKA envisions an infrastructure of public terminals or displays in the hospital in addition to the patients' personal devices[5]. It is necessary that the personal health information which can be processed by an application on a public terminal or display cannot be misused. Misuse can take two forms: alteration of the stored information by unauthorized parties, and privacy of the patient's health history. Potential solutions to the first problem were proposed by introducing different types of acces control models, see e.g. [4] and [5]. Therefore, this paper focuses on the second problem: how to make sure that the patient's information is not misused. The example which will be used throughout this paper is a public terminal with a browser that allows viewing of the information stored on the patient's smartcard and on file in the hospital database (cf. Figure 1). The public terminal or display has to forget the content and the actions performed after the patient ejects her smartcard, leaving no information about her health history in the browser cache. The same applies to a printer which may have been used during the session. In general, a service or an application is used, which may not be in the patient's or hospital's administration or trust[6] domain, therefore it is uncertain that sensitive personal data are treated in the expected way. The public terminal in the untrusted zone[7] enables access to files

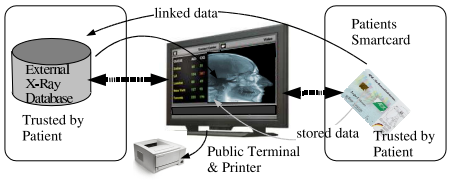

Fig. 1. Architectural Overview

[3] We would like to point out that these issues are not limited to the hospital environment and also appear in other areas, for instance in e-commerce and web-services in general.

[4] Smartcards like this are currently being specified and will be used in the near future in the German health system under the name "Krankenkarte".

[5] While every patient will be supplied with a smartcard, not every person will own a PDA.

[6] Here, trust is defined as the patient being confident that her data is not misused.

on the trusted smartcard and access to linked external information, for instance, X-ray images. The smartcard is viewed as trusted because it is owned by the user. The external database is maintained by the hospital and is therefore also trusted. The terminal communicates directly with the smartcard and external sources.

2 Attacker Model

The aim of an attacker in this scenario would be to gain access to private health information. The attacker may gain control over some of the software on the public terminal, or gain complete control over the terminal after the user left it. He may read and insert communication between the smartcard and the terminal, or read and insert communication between the terminal and the backend database. In addition, the attacker may also introduce a fake terminal. An attack that involves an attacker looking at the display of the terminal is not considered.

3 The Approach

Our approach follows closely the idea presented by Korba and Kenny in [6] for solving the problem that a user can keep control over transmitted personal data is based on the following observation: the interests a service or application user has in dealing with sensitive data are similar to those of providers of copyrighted digital contents. Both, the copyrighted content provider and the patient, i.e., the personal data provider, are interested in making the supplied data available only for limited use and processing. Furthermore, unauthorized onward transmission and use should be prevented. Subsequently, control over transmitted data or contents has to be enforced.

This parallelism of interests between content providers and patients (service users) with regard to the processing of data makes digital rights management systems a suitable toolset for the protection of sensitive personal data. Personal data is sent in a DRM-like protected way to the service-providing device preventing unauthorized usage and information leakage. Sensitive personal data has a license attached to it when communicated to the service providers. The license limits the use of this personal data. The service user now takes the role of content provider and license issuer. Because it would be unmanageable if every patient had her own slightly different license attached to her data, patient interest groups should act as liaison and offer standardised licenses.

This is a contrary approach to classical anonymization techniques with the concepts of data minimality and data confusion, because a technical implemented temporal extension of the domain of trust is used.

[7] The terminal is considered untrusted because it is easy to tamper with given its public location, while it is very hard to administrate it such that it remains tamper resistant.

4 Technical Solution

Successful deployment of a DRM system requires a component processing DRM content. This component can ensure that the applications which are executed are untampered with and provide a safe execution environment. The Trusted Computing Group [7] is developing extensions to computing platforms to ensure this. Because major industry players, including hardware and software manufactures and content providers, are involved in specifying this platform one can assume that DRM capabilities will become pervasive. The TCG platform can produce signed attestations of the integrity of the software.

Technically the TCG specifies hardware extensions by which different stages of starting and running a platform can be verified by measurement functions and reported to the TPM. By this, the trusted domain is extended with every successful verified component (BIOS, firmware of devices, bootloader, operating system). This extension of trust is illustrated in figure 2. If the platform has successfully started and all the hash values of the measured components matches the expected values of a known state platform, the TPM unlocks signing functions to be able to prove its known state. Microsoft proposes an operating system with

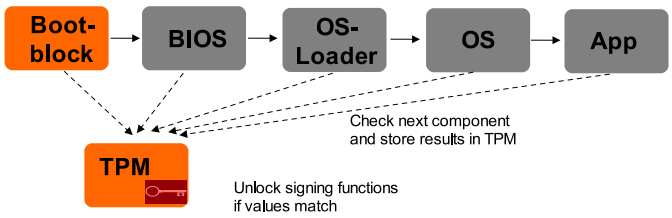

Fig. 2. Boot procedure with code verification

the so called Next Generation Secure Computing Base NGSCB [8] which extends the existing context in which a process can be executed with a secure context environment. Only verified code can be executed in this protected context. Debugging or getting out memory is not possible and should be supported by a special processor mode in future. ARM the well known microprocessor designer as well proposes a model [9] with a couple of similarities, especially the division of context in a normal side and a secure side.

To decide if the platform which should process sensitive personal data behaves as it claims to, one has to know about the software and the platform. Trusted Computing mechanisms can guarantee a proper and verified execution. But it will be hard to know about all soft- and hardwarecomponents and about different versions of them. This makes a third party necessary to classify software and hardware components as trustworthy or possible to build trustworthy platform on it.

4.1 The implementation

In our proof of concept implementation a trustworthy platform e.g. with TCG compliant TPM wasn't available. This means the core root of trust cannot be the TPM chip. Instead we treated the used operating system with code integrity checking functionality as the core root of trust and the information about executed software on this system are reliable. We also excluded the use of a third party software component evaluator. The user, respective the users device knows how the terminal has to look like. A secure execution context comparable to NGSCB was also not available. To simulate the attribute of obliviousness (after the terminal was used, it should forget about everything) the application with the user data is executed from a ramdisk which is reformatted after the usage. To allow rapid protoyping, the smartcard functionality was implemented on a PDA. In figure 3 the interaction between the smartcard simulating PDA and the terminal is shown.

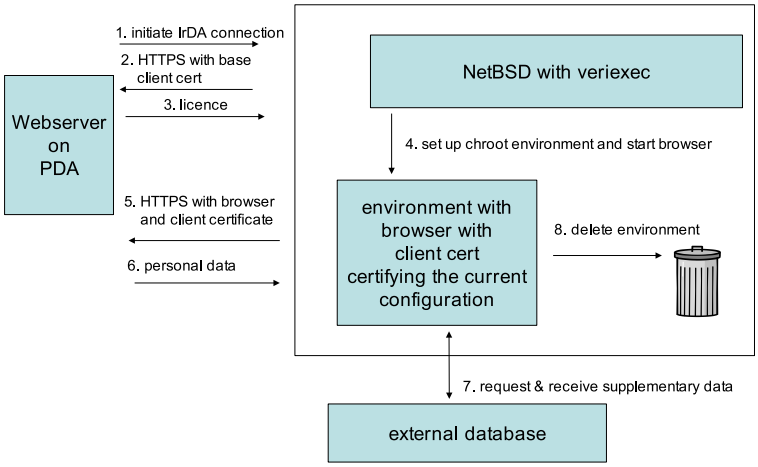

Fig. 3. Phases during the use of the terminal

For the hospital environment, we extend the functionality of the smartcard with the capability of verifying these attestations and, thus checking the integrity of the public terminal or display. The current implementation of the public terminal is based on the NetBSD operating system [10] with the so called VERIEXEC option which supports the execution of applications with a valid hash signature only.

The public terminal runs a trusted daemon waiting for events on the IrDA-port (using userland birda[11] implementation) to interact (step 1 in figure 3) with the PDA used to emulate the smartcard. An IP connection between the

terminal and the PDA is established. The daemon establishes a secure HTTP connection using a client certificate that is signed by a root trusted by the PDA (step 2). Through this connection, the public terminal gets the licence for using the data stored on the PDA. The licence contains a list of access rights which in this implementation is either *view* or *view and print* (step 3). Based on these rights, the daemon sets up a chroot environment on a ramdisk (step 4) with or without a printer device. A browser with yet another client certificate attesting the environment that has been set up is launched and connects via HTTPS to the server on the PDA (step 5). This browser is now permitted to access the personal data (step 6). The personal data may contain links to external documents, like X-ray images (step 7). These hyperlinks are HTTPS hyperlinks with embedded login information to the external patient information database. The daemon that set up the environment continually polls the PDA to find out if there still is a contact. If the contact is lost for more than five seconds, the ramdisk is deleted and, thus, no trace of the personal data left on the device (step 8).

5 Discussion

The implementation represents a first step towards using DRM-like mechanisms to protect the privacy of users of public terminals. The used operating system supports a verified execution but in itself can not represent the same core root of trust as trusted computing hardware. The PDA can issue the right to view and print. Printing is a digital transfer of sensitive data to another device, the printer. This means that the printer itself should have to respect the terms of the licence. Currently, a printer without permanent storage is used.

The implementation described in the previous section does not address the threat that the browser may be tricked into posting sensitive information to untrusted sites. To this end, further isolation of the network environment is required, similar to the isolation of the filesystem provided by the chrooted ramdisk.

The use of stunnel[12] and HTTPS is very computation intensive for the user's device. Using NGSCB-like DRM mechanisms could reduce this load and lead to a solution closer to the capabilities of a real smartcard.

6 Related Work

There is some work that is related to the approach presented here. As stated before, the idea of using DRM like mechanisms for protection of personal data was discussed by Korba and Kenny [6]. However, they did not present a working system architecture or proof of concept implementation. Bussard et al. [13] demonstrate how to diplay sensitve information in federations of devices. However, their approach doesn't work if the information is too complex to be displayed on a limited screen (e.g. x-ray pictures). Kohl [4] pointed out that privacy is in fact a big issue in a hospital environment, but assumed a central organization for data storage and processing. Privacy through the use of identity management in a

mobile computing environment is proposed in [14]. It is based on the retention of personal data and can not be controlled once they are given in foreign hands. Agrawal et. al [15] attach a licence to data in a database. This approach is a good way of ensuring privacy as long as the data does not cross administrative domain boundaries.

Closer to the method presented here is the suggestion of Langheinrich in [16]. A policy is attached to personal data to create a sense of accountability. The approach of Mont et al. in [17] uses a third party to trace and audit the use of personal information.

7 Conclusions and Future Work

The results from the first trials are encouraging and lead us to believe that DRM can be used to enforce privacy. The setting in the hospital is almost ideal for DRM. It can be expected that only few different companies will provide equipment for the hospitals. Hospitals are highly regulated and, therefore, there is interest by the hospital to ensure privacy. Additionally, this approach can be used to shift the work of ensuring the correct handling of data from the person installing and maintaining the pervasive computing environment to the software vendor for the viewer of the data.

Future work includes a port of the current implementation to NGSCB and a closer look at certificate management and revocation. In addition, different DRM systems approaches have to be evaluated to find out which one supports the need of handling of personal data. It will also be interesting to implement the certificate validation on a smartcard to verify the performance.

References

1. Weiser, M.: The Computer of the 21st Century (1991) Scientific American, vo.265. no.3, Sept.1991, pp 66-75.
2. Iachello, G., Abowd, G.D.: Security requirements for environmental sensing technology. (2003) 2nd Workshop on Ubicomp Security, Oct. 2003, Seattle, WA, USA.
3. Müller, G., Kreutzer, M., Strasser, M., et al: Geduldige Technologie für ungeduldige Patienten, führt Ubiquitous Computing zu mehr Selbstbestimmung? In: Total vernetzt. Springer: Berlin, Heidelberg, New York. (2003) 159–186
4. Kohl, U.: From Social Requirements to Technical Solutions - Bridging the Gap with User-Oriented Data Security. In: Proceedings IFIP/Sec '95, Cape Town, South Africa, 9-12 May. (1995)
5. Brose, G., Koch, M., Löhr, K.P.: Entwicklung und Verwaltung von Zugriffsschutz in verteilten Objektsystemen - eine Krankenhausfallstudie. (2003) Praxis der Informationsverarbeitung und Kommunikation (PIK).
6. Korba, L., Kenny, S.: Towards Meeting the Privacy Challenge: Adapting DRM. (2002) ACM Workshop on Digital Rights Management.
7. Trusted Computing Group: TCG Backgrounder. (2003)
8. Microsoft Corporation: NGSCB: Trusted Computing Base and Software Authentication. (2003)

9. ARM: TrustZone Technology - Secure extension to the ARM architecture at http://www.arm.com/products/cpus/arch-trustzone.html. (2004)
10. NetBSD: http://www.netbsd.org. (2004)
11. birda: birda package at http://www.netbsd.org. (2004)
12. stunnel: www.stunnel.org. (2004)
13. Bussard, L., Roudier, Y., Kilian-Kehr, R., Crosta, S.: Trust and Authorization in Pervasive B2E Scenarios. In: Proceedings of the 6th Information Security Conference (ISC'03) Bristol, United Kingdom, October 1st-3rd. (2003)
14. Jendricke, U., Kreutzer, M., Zugenmaier, A.: Mobile Identity Management. Technical Report 178, Institut für Informatik, Universität Freiburg (2002) Workshop on Security in Ubiquitous Computing, UBICOMP.
15. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic Databases. In: 28th Int'l Conf. on Very Large Databases (VLDB), Hong Kong. (2002)
16. Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments. (2001)
17. Mont, M., Pearson, S., Bramhall, P.: Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services. (2003) HPL-2003-49.

Part of this work was funded by the DFG / Gottlieb Daimler and Carl Benz Foundation.