Securing the E-Health Cloud

Hans Löhr
Horst Görtz Institute
for IT Security
Ruhr-University Bochum
Germany
hans.loehr@trust.rub.de

Ahmad-Reza Sadeghi
Horst Görtz Institute
for IT Security
Ruhr-University Bochum
Germany
ahmad.sadeghi@trust.rub.de

Marcel Winandy
Horst Görtz Institute
for IT Security
Ruhr-University Bochum
Germany
marcel.winandy@trust.rub.de

ABSTRACT

Modern information technology is increasingly used in healthcare with the goal to improve and enhance medical services and to reduce costs. In this context, the outsourcing of computation and storage resources to general IT providers (cloud computing) has become very appealing. E-health clouds offer new possibilities, such as easy and ubiquitous access to medical data, and opportunities for new business models. However, they also bear new risks and raise challenges with respect to security and privacy aspects.

In this paper, we point out several shortcomings of current e-health solutions and standards, particularly they do not address the client platform security, which is a crucial aspect for the overall security of e-health systems. To fill this gap, we present a security architecture for establishing privacy domains in e-health infrastructures. Our solution provides client platform security and appropriately combines this with network security concepts. Moreover, we discuss further open problems and research challenges on security, privacy and usability of e-health cloud systems.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*information flow controls, security kernels*; J.3 [**Life and Medical Sciences**]: Medical Information Systems

General Terms

Security

Keywords

E-Health, security architecture, information flow, isolation, client platform security

1. INTRODUCTION

The application of information technology to healthcare (healthcare IT) has become increasingly important in many countries in the recent years. There are continuing efforts on national and international standardization for interoperability and data exchange. Many different application scenarios are envisaged in electronic healthcare (e-health), e.g., electronic health records [12, 23, 22], accounting and billing [17, 24], medical research, and trading intellectual property [15]. In particular e-health systems like electronic health records (EHRs) are believed to decrease costs in healthcare (e.g., avoiding expensive double diagnoses, or repetitive drug administration) and to improve personal health management in general.

Examples of national activities are the e-health approach in Austria [23], the German electronic Health Card (eHC) system [12] under development, or the Taiwan Electronic Medical Record Template (TMT) [22]. In Germany each insured person will get a smartcard that not only contains administrative information (name, health insurance company), but also can be used to access and store medical data like electronic prescriptions, emergency information like blood group, medication history, and electronic health records. The smartcard contains cryptographic keys and functions to identify the patient and to encrypt sensitive data. The TMT in Taiwan concentrates on a standardized document data structure to ease information sharing, but also contains a similar infrastructure based on smartcards allowing to share and transfer EHRs. A common approach in all these systems is to store medical data in central data centers, which build the core concept of a centrally managed healthcare telematics infrastructure.

On the international basis the ISO (Technical Committee 215) [16] and the Health Level 7 consortium (HL7) [14] define standards for e-health infrastructures. While they also include specifications for security and privacy aspects, their main focus is currently the interoperability and definition of common document exchange formats and nomenclature of medical data objects.

Obviously e-health systems store and process very sensitive data and should have a proper security and privacy framework and mechanisms since the disclosure of health data may have severe (social) consequences especially for patients. For example, banks or employers could refuse a loan or a job if the data about the health of a person is available. If health data is leaked outside the system deliberately or accidentally, the responsible health professionals or IT providers would have to face severe legal penalties for violating privacy laws.

When addressing privacy regulations with technical solutions, we are faced with a number of difficulties: E-Health

systems must accommodate various work flows, not only related to the patients' medical data, but also accounting and billing of treatments, medication, etc. Moreover, for smartcard-based solutions, the system must ensure that there is some way to access medical data (which might be life-critical in some situations) even if the owner of the smartcard is unable to authenticate to the system, e.g., because he or she is unconscious. In other situations, data must be accessed when the smartcard owner is not present, e.g., in case a relative buys medication for the patient at a pharmacy. Addressing such issues in an appropriate way presents a major challenge for research and industry. In particular, we can observe that current e-health solutions and standards mainly focus on network security and access control policies, however, they do not address the client platform security appropriately [26], i.e., the security of the software and hardware that is used by health professionals locally.

Contribution and Outline.

In this paper, we discuss the general problems of e-health systems and provide a technical solution for the protection of privacy-sensitive data, which has not been appropriately addressed yet for end-user systems. In particular, our contributions are as follows:

- We describe an abstract model of e-health clouds (Section 2), which comprehends the common entities of healthcare telematics infrastructures. Based on this model, we outline three main problem areas for security and privacy (Section 3), namely (i) data storage and processing, (ii) management of e-health infrastructures, and (iii) usability aspects of end-users.

- We present a security architecture for privacy domains in e-health systems (Section 4) which leverages on modern security technology of commodity platforms. This architecture extends the protection of privacy-sensitive data from centrally managed secure networks to the client platforms of the end-users. For each application area a separate privacy domain is established and it is enforced both centrally and locally on each platform.

Our solution presents results from some ongoing research and development e-health projects where our results cover the problem areas (i) and partially (ii). We also discuss the remaining research problems (Section 5).

2. MODEL OF THE E-HEALTH CLOUD

This section gives an overview of typical e-health infrastructures as they are available as products or planned to be deployed in national healthcare information technology projects. We present an abstract model of the resulting e-health clouds.

In the past, health care providers (such as the family doctor) have stored medical records of their patients on paper locally. This allowed a controlled environment with easy management of data privacy and security: keeping the paper records in a locked cabin at the doctor's practice. Even the increasing use of personal computers and modern information technology in medical institutions allowed for a moderate effort to manage privacy and confidentiality of individual medical records. This was due to the decentralized and locally managed infrastructure of each institution.

But nowadays outsourcing of IT infrastructure (e.g., cloud computing) and other services (e.g., billing processing and accounting for medical practices) leads to a complex system where privacy-sensitive data are stored and processed at many different places. Hence, it becomes attractive to store and process healthcare data "in the cloud" (at outsourced data providers that can be accessed via the Internet). While such e-health systems promise a more cost-efficient service and improved service quality, the complexity to manage data security and privacy increases, too.

In order to identify and discuss the different problems areas, we present first a simple model and then extend it to an advanced model of the "e-health cloud". We identify the involved parties and main components that are relevant for the focus of our paper.

Terminology.

Throughout this paper we use the following terms:

Health professional: person who delivers health care services, e.g., physician, dentist, pharmacists, etc.

Health care provider: organization that provides services of health professionals, e.g., doctor's practice or hospital.

Personal Health Record (PHR): database of medical data objects and health-related data managed by a patient.

Electronic Health Record (EHR): database of medical data objects and health-related data managed by health professionals.

Note that sometimes the separation of PHR and EHR is not made clearly in the literature. But due to different legal implications in certain countries this distinction is important.

Simple Model of the E-Health Cloud.

We first consider a simple model that underlies commercial systems like Google Health[1], Microsoft HealthVault[2], and ICW LifeSensor[3]. In these systems patients store their own health-related data on certain web servers: the so-called Personal Health Record (PHR). In this model, patients track, collect, and manage the information about their health at online web sites. They can enter dates and periods of sickness, their appointments with doctors, and any other data related to their health. Patients can also import data in their PHRs they get from health professionals, such as x-ray photos or laboratory tests from their family doctor or dentist. Figure 1 illustrates this model and shows the involved parties.

The PHRs are stored on a server of a third party in the cloud. The PHR server provider is responsible for ensuring data protection. Typically, patients can define role-based access rights for individual health professionals. For example, they can define full access to their family doctor, but only restricted access to some data to their fitness trainer or health coach. The advantages of such an approach are that the PHR is accessible from everywhere because of the centralized management (IT outsourcing). The patient can easily give one doctor access to data and test results that were determined by another doctor, when the data is stored

[1] https://www.google.com/health/
[2] http://www.healthvault.com/
[3] https://www.lifesensor.com

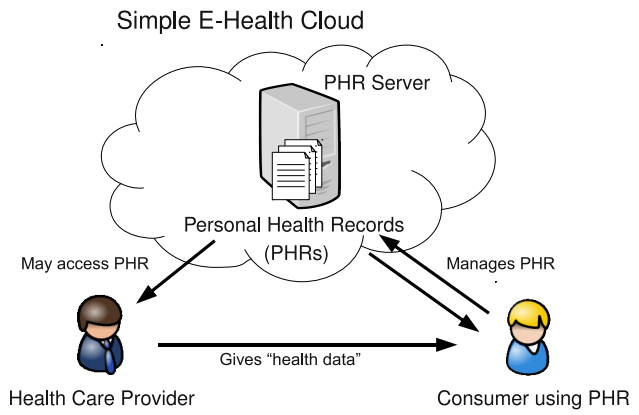

Figure 1: Simple E-Health Cloud model. Patients manage their own personal health records.

in the PHR. This can help to avoid double examination. Moreover, due to the individual management of PHRs by the patients, it is expected that people are more aware of their own health. This could reduce the healthcare costs in the long term as well. However, from a technical perspective this model has a great disadvantage regarding patients' privacy. On the one hand, patients need to manage complex access rights and need to understand their implications. On the other hand, they need to rely on the robustness and correctness of the security mechanisms implemented at the PHR server provider. In general, it may be possible for the server provider to gain access to the data stored in PHRs.

Advanced E-Health Cloud Infrastructure.

In contrast to PHRs, which are managed by the patients, Electronic Health Records (EHR) are managed by health professionals only. In most countries this involves different legal requirements and a clear distinction between PHRs and EHRs. As a result, infrastructures that involve EHRs are usually more complex than our simple e-health cloud model. Figure 2 shows the advanced model, which not only involves more parties (e.g., health insurances), but also includes some technical means to enforce data security and privacy of EHRs.

The general requirement in this model is still the functional and semantic interoperability of the data stored in EHRs. The EHRs are created, maintained, and managed by health care providers, and can be shared (via the central EHR server in the cloud) with other health professionals.

But storing and processing EHRs is not the only service that can be outsourced to the cloud. The health care providers can use billing services that manage their billing and accounting with the health insurances of the patients. This is a typical scenario that can be found in practice: Many doctors outsource the billing to third party providers. Those billing services accumulate the billing of several patients for different health insurances, but also for various health care providers at the same time. As a consequence, privacy becomes an even more important aspect in this model because health insurances or billing services should not be able to access private details of EHRs.

To protect the EHR data, smartcards are typically used to (1) authenticate health professionals and patients, (2) sign EHR documents to provide authenticity, (3) encrypt the EHR data before they are stored in the cloud, and (4) authorize the access to EHR data. Data and services of the e-health cloud can only be accessed with special interface connections to the telematics infrastructure boundary. This interface connection is typically a special hardware device that establishes secure network connections via a Virtual Private Network (VPN) to the e-health data centers. Due to the increased privacy requirements, many countries define standards and specifications for national e-health infrastructures that include technical means for security and privacy.

However, existing security concepts in e-health concentrate on controlling access to data (e.g., smartcard-based access control to web-based PHRs and EHRs), protection of data transfer (encryption for confidentiality, digital signatures for integrity and authenticity), and network security (firewalls, VPNs). The latter focuses on the separation of different networks, e.g., administrative networks of health insurances from EHR servers and from other applications. However, little care is taken on what happens after access to data is allowed, i.e., how data is processed and stored on end-user client platforms. Viruses or Trojan Horse programs can corrupt data and eavesdrop on patients records, violating both legal and individual privacy requirements.

Example: The German electronic Health Card (eHC) system [12, 10] under development defines that in the compulsory health insurance system, each patient has an eHC smartcard. The eHC is mainly used for storing administrative data (for billing with the health insurance), but also includes functionality to encrypt medical records that are going to be stored on EHR servers, and to authorize access to EHR data. When a medical doctor wants to upload or download EHR data of a patient, this patient has to provide his/her eHC and to enter a PIN in order to initiate encryption (upload) or to authorize access (download). Moreover, medical doctors have their own smartcard, the Health Professional Card (HPC), which is used to digitally sign documents that are stored in a patient's EHR, and to authenticate themselves as legitimate medical personnel. Each health care provider has to have a special smartcard reader where the eHC and the HPC are inserted whenever access to the EHR is requested. A special *connector* locally interconnects the computing platforms of the health care provider with the smartcard reader and the telematics infrastructure. The connector is also used to connect to other networks that provide additional applications, but which are not part of the telematics system itself [11]. The client platforms and the local networks of health care providers are out of scope of the healthcare telematics security requirements. In addition, when patients want to administer their personal data or manage access rights, they also need to use corresponding client platform systems. In both cases it is completely up to the end-users to secure their systems appropriately. Thus, the software on these computer systems can be identified to be the most likely attack target [25], as they are standard PC systems with commodity operating systems that offer standard services, e.g., e-mail and Internet access.

Other countries in Europe (e.g, Austria [23]) or Asia (e.g., Taiwan [22]) plan similar architectures.

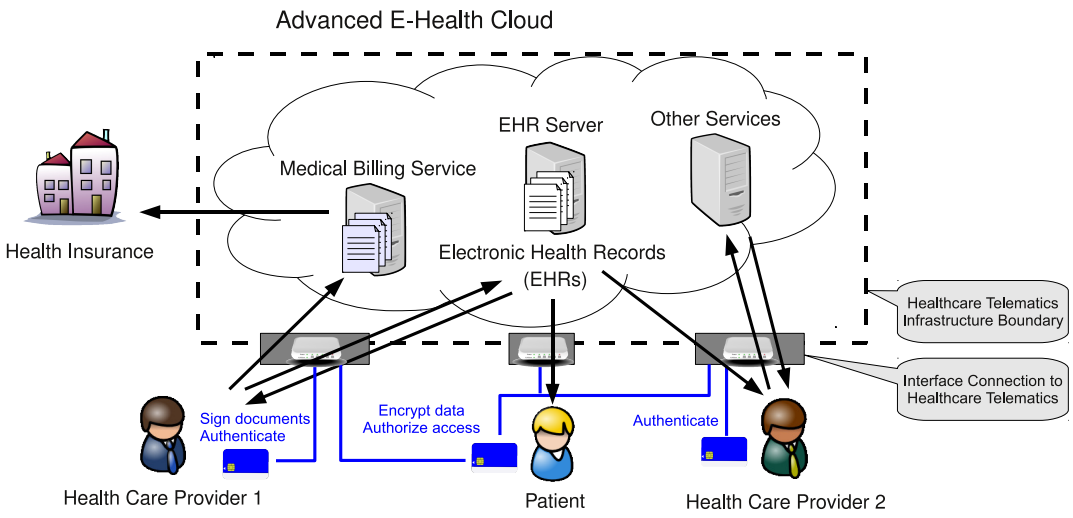

Figure 2: Advanced E-Health Cloud model. Health professionals manage health records of patients.

3. PROBLEMS OF E-HEALTH CLOUDS

In this section, we give a systematic overview of the threats in the privacy-sensitive context of e-health clouds. The processing of healthcare data of patients has technical, but also legal problems that one has to deal with. In this paper, we focus on the technical aspects. We therefore analyze three different problem areas.

3.1 Data Storage and Processing

Security and privacy issues exist where the medical data of the health records are stored and processed, i.e., at the PHR or EHR server and, of course, at the local computer infrastructure of health care providers. Access control mechanisms and data encryption can ensure confidentiality of the medical data, and great efforts are done in this direction in many specifications, such as the German eHC [12], and standardizations, such as HL7 and ISO/TC 215 [16].

Data Centers.

Storing privacy-sensitive data in central data centers bears the risk of information leakage to unauthorized entities. Sensitive data must be sufficiently protected, e.g., by means of strong cryptographic encryption. Moreover, it must be possible to administer and maintain the data center without letting administrators gain access to patient data.

Client Platforms.

The security of end-user systems is another problem that is rarely dealt with. Most specifications that we are aware of define this as "out of scope". End-user systems are the PCs and network infrastructure at the doctor's practice or the computing platforms of information systems in hospitals. Especially, medical doctors who run their own small practice do usually not have the competence and time to professionally manage their IT systems to be sufficiently protected against malware threats. On the other hand, they use their computer systems not only for accessing health records of their patients, but also for other applications, such as billing systems, or Internet browser. But today's commodity operating systems that are used do not offer sophisticated security mechanisms nor are they implemented in a robust way as high-assurance systems. Due to architectural limitations they do not offer sufficient runtime protection of applications and operating system software, they lack information flow control mechanisms and secure user interfaces. All this makes these systems vulnerable to malware attacks that could steal passwords and secret data, or leak privacy-sensitive data to illegitimate destinations on the Internet.

Mobile Storage Devices.

Moreover, those computer systems are usually used by several persons, e.g., medical assistants, and they may connect them with mobile storage devices, such as USB memory sticks, for transferring data to other platforms. Data that is transferred in this way usually leaves the control of any security mechanisms of the e-health infrastructure.

3.2 Management of E-Health Infrastructure

On a larger scale, the whole infrastructure of an e-health cloud has several risks that threaten the privacy of health data. Both medical and administrative data of patients are processed at several places in the e-health cloud, and the usage of smartcards and access control mechanisms alone does not provide the necessary protection.

Cryptographic Key Management.

Complex infrastructures must be managed and this comprises additional security and privacy issues. The usage of encryption requires management of cryptographic keys, smartcards must be personalized and issued to their users. One question that is often insufficiently answered in this context concerns who is in control of the cryptographic keys. A naive approach would say the patient of course. But how to handle lost or stolen cards when the encryption keys are lost as well? Do the card issuer or the EHR server have backup copies of the keys? But backup strategies must also take into account the privacy requirements of health data. For example, in many European countries, and especially in Germany, it is required by law that the patients themselves have the full data sovereignty over their health data. This

means no other party is allowed to circumvent privacy decisions and access rights definitions of the patient regarding EHR data. But if the card issuer or even the EHR server providers maintain backup copies of the cryptographic keys for reasons of issuing backup smartcards in case of theft or loss, they could in principle decrypt and access the EHR data directly.

Management of Certificates.
As in any public key infrastructure, certificates must be managed to ensure authenticity of key holders (smartcards, connectors, server, etc.). This includes issuing and distributing certificates as well as updating revocation lists.

Management of Hardware/Software Components.
Besides the cryptographic infrastructure, other components must be managed and maintained as well. This includes the hardware and software components that are used at EHR servers, billing servers, and computing devices of health care providers. Security-critical components, such as smartcard readers or connectors to protected networks, should be certified and tested properly. The installation and update of software components requires a secure distribution mechanism. On the one hand, it must be possible to allow changes in software configuration due to legitimate updates. On the other hand, unauthorized and malicious changes (e.g., due to malware attacks), must be detectable to stop further usage or to exclude the infected components from the e-health infrastructure.

3.3 Usability and User Experience

Finally, our third problem area is concerned with the end users, i.e., the health professionals and the patients. If security controls and configurations are too complicated, ordinary people would not be able to use them or would try to ignore or circumvent them. For example, remembering a PIN for the smartcard may be too hard for older patients. People tend to write the PIN on paper or even on the smartcard in these cases, which renders the security aspect of having the PIN at all useless.

From the perspective of health professionals, there are other issues. As mentioned before, doctors are not IT professionals and they might be overstrained with the configuration and secure setup of all the software components. Moreover, IT-related tasks that delay their own (medical) processes will disturb them and they will tend to ignore or circumvent them. For example, inserting smartcards and entering PINs in a smartcard reader whenever they want to access an EHR might be too time consuming — or even impossible in case a patient wants to consult his/her doctor via telephone.

4. SECURE E-HEALTH INFRASTRUCTURE

The problem areas above show that e-health clouds impose a variety of security and privacy risks. Ideally, all of them should be solved technically and transparently for the users. In the following we present a technical solution to address particularly the end-user platform security issue. Compared to other efforts, especially national and international standardizations, this topic is not addressed sufficiently.

We propose to base a secure e-health infrastructure on Trusted Virtual Domains (TVDs) to ensure fundamental security and privacy properties. In this section, we first introduce privacy domains for healthcare systems. Then we discuss our realization based on a security kernel and TVDs.

4.1 Privacy Domains for E-Health

In the context of e-health, privacy protection of the patients' data is a primary concern. Technological solutions should be employed to support legal and contractual regulations.

We propose to construct *privacy domains* for the patients' medical data as a technical measure to support the enforcement of privacy and data protection policies: Systems (e.g., a client PC) must be able to partition execution environments for applications into separate domains that are isolated from each other. Data is kept within a privacy domain, and the domain infrastructure ensures that only authorized entities can join this domain. Moreover, data leakage from the domain is prevented by the security architecture and the domain infrastructure. Therefore, the same system can be used for different work flows that are strictly isolated. Figure 3 illustrates the privacy domains applied to our e-health cloud model.

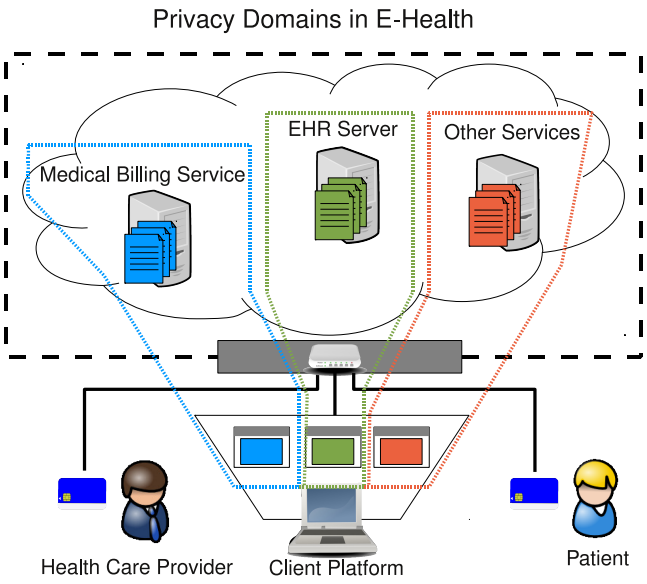

Figure 3: Privacy Domains in the E-Health Cloud. For each application a privacy domain is established in the cloud and also enforced on the client platforms of the health care providers.

An important aspect for the deployment of any new infrastructure in practice is the integration of legacy systems. With our concept of privacy domains, it is possible to re-use existing applications running on a legacy operating system within a privacy domain. Furthermore, data import into the domain can be accomplished via gateways and filters to connect the privacy domain infrastructure to legacy systems.

Application Scenario.
As an example, consider a doctor who wants to use an accounting software to submit bills to health insurances via a

dedicated healthcare network[4], and another system to store and process patients' medical data. In addition, the doctor needs web access and must be able to send and receive e-mails.

These different work flows should be separated from each other: The health insurance should not get access to the detailed medical data, and security problems arising from Internet access and vulnerabilities in the web browser should not influence the accounting process, or have impact on the medical data. Only the correct accounting software may connect to the healthcare network. However, it might be desirable for the doctor to be able to send medical data from an EHR concerning a particular case to a specialized colleague or healthcare organization using the normal e-mail client – but without risking the disclosure of such data to malware that might have infected the e-mail software, or to attackers in the Internet. The accounting software or applications processing the medical data might require specific (perhaps outdated) operating systems (e.g., Windows XP), whereas the web browser (and the operating system on which it runs) should always be updated to include the latest security patches.

To achieve these objectives, three privacy domains with different requirements are used. One privacy domain, the accounting domain, is restricted to software authorized to access the accounting network. The doctor uses a virtual machine which is part of this TVD and runs the accounting software. A second privacy domain, the e-health domain, is dedicated to the storage and processing of EHRs. The doctor runs software in this TVD to access a patient's medical data. A third domain – which is neither part of the accounting domain, nor of the e-health domain – contains untrusted programs such as a web browser (e.g., Firefox) and e-mail client (e.g., MS Outlook). Only this VM is allowed to access the Internet without restrictions; the accounting software is restricted to connect to accounting servers, software in the e-health domain is only allowed to connect to relevant e-health servers. Its software, including the operating system, can be updated independently from the other domains. The graphical user interface shows the different domains framed in different colors to help the user distinguish them from each other.

When medical data is stored on external storage (e.g., a USB disk) or transferred to the e-mail client via copy-and-paste, the system automatically encrypts the data with a key that is accessible only in the corresponding privacy domain. The encrypted data can be moved to another machine (either by physically transporting a USB disk, or by sending it via the Internet). When it reaches the correct domain again, the system on the target platform decrypts the data. Encryption and decryption is completely transparent to users – they will only notice that the data can only be read properly with applications executing in the correct privacy domain.

To export data to legacy systems a special gateway is introduced. This is necessary, for instance, when medical data have to be accessed by some doctor or hospital that is not (yet) connected to the privacy domain. A dedicated gateway allows for better control of the data. If data export is only possible via a dedicated gateway, unintentional disclosure of sensitive data can be prevented.

[4] In Germany, there already exists such a network, called KV-SafeNet [17], which is already used by many doctors and healthcare institutions.

4.2 Our Realization using TVDs

The major goals of our project include the separation of medical data from other data such as billing and accounting, as well as the integration of e-health cards into the system. Currently, a working prototype of the basic technology exists, and we are working on the user interface and a usable implementation for end users. We plan to conduct a user study with approximately 250 users over a period of 18 months, beginning next year.

In the following, we describe the basic technology of our proposal. Our realization is based on Trusted Virtual Domains.

Trusted Virtual Domains.

Trusted Virtual Domains (TVDs) [13, 3, 4] have been developed in recent years as a security framework for distributed multi-domain environments which leverages virtualization and trusted computing technologies. Although in the beginning primarily proposed for use in large-scale data centers [2], TVDs can also be useful in other scenarios. In this section, we give a brief overview of the TVD concept and its features.

In a virtualized environment, virtual machines (VMs) that share the same physical infrastructure execute operating systems with different applications and services. Each virtual machine runs in a logically isolated execution environment (which we call *compartment*), controlled by an underlying security kernel that acts as virtual machine monitor (VMM). The user's work space is now executed by a virtual machine that is hosted by the security kernel running on the physical platform along with other architectural components.

A TVD is a coalition of virtual machines that trust each other, share a common security policy and enforce it independently of the particular platform they are running on. Moreover, the TVD infrastructure contains the security kernel and the physical components on which the VMs rely to enforce the policy. In particular, the main security features of TVDs and the TVD infrastructure are:

- *Isolated compartments.* The security kernel provides containment boundaries to compartments from different TVDs, allowing the execution of several different TVDs on the same physical platform.
- *Trust relationships.* A TVD policy defines which platforms (including the security kernel) and which VMs are allowed to join the TVD. For example, platforms with their security kernel, as well as individual virtual machines, can be identified via integrity measurements taken during their start-up.
- *Transparent policy enforcement.* The security kernel enforces the security policy independently of the compartments.
- *Secure communication.* VMs belonging to the same TVD are connected through a virtual network that can span over different platforms and that is strictly isolated by the virtual networks of other TVDs.

To provide these features, TVDs use state-of-the-art security technology: During system operation, the security kernel has to guarantee the strict isolation of different domains. Whenever data leaves a domain, e.g., during transfer over a network or when storing data on a disk, security services

encrypt all data with a cryptographic key that is available only in the corresponding TVD. In this manner, TVDs prevent accidental information disclosure and help to thwart malware attacks. Trusted Computing technology is used to protect data from attacks while the system is powered off[5] and to verify the integrity of platforms before they are allowed to join a TVD.

A major feature of the TVD infrastructure is its automatic management. TVD establishment, key management, and policy enforcement are completely transparent to users. The infrastructure verifies the integrity of client platforms when they join a TVD and distributes keys and policies. On the client platform, policy enforcement and data encryption are handled by a security kernel without any user interaction. As long as users do not try to violate the policy, the only difference compared to a conventional system they notice are visual indicators of the domain they are working in. In our implementation, this is a colored bar at the top of the screen, e.g., green when the user is working in the medical domain and blue for accounting.

To secure the connection between different platforms in a TVD, we use an IPSec-secured virtual private network.

Client Platform Security.

Figure 4 shows the structure of a TVD with a client platform and a TVD master (a server for the management of the TVD infrastructure) for each domain. On the client platform, a security kernel is running on top of the hardware, providing isolated virtual machines for applications and conventional operating systems. Moreover, there is a TVD proxy for each TVD, which manages the TVD on the client and configures the security kernel according to the TVD policy. A secure graphical user interface (secure GUI) provides input and output, ensuring that users can always reliably identify with which compartment they are interacting. The secure GUI also prevents other compartments from reading user input when they are not authorized to do so.

Furthermore, the client platform contains a trusted hardware component which can be used for the verification of the integrity of the software on the client (in particular, the security kernel). The most widespread trusted hardware component is the Trusted Platform Module (TPM) [27]. Currently, it is usually implemented as a separate chip integrated on the mainboard of a computer. Many computers (including almost all business notebooks sold today) are equipped with TPMs, although this is usually not advertised explicitly.

TPMs provide a set of security and cryptographic functions, such as public key encryption, digital signatures, secure key storage, non-volatile memory, etc. For TVDs, three functionalities are especially important:

- *Authenticated boot* [21]: When the system starts, the platform computes cryptographic hash values (which can be considered like a "secure fingerprint") of the components that are loaded and executed (e.g., firmware, boot loader, operating system kernel). These values are securely stored inside special-purpose registers of the TPM, the platform configuration registers (PCRs).

- *Trusted storage* [21]: The use of cryptographic keys created by the TPM can be restricted to specific PCR values. This implies that these keys can only be used when the correct system has been started.

- *Attestation* [27, 9]: The TPM can use special-purpose cryptographic keys, called attestation identity keys, to sign the current PCR values. This attestation mechanism can be used in cryptographic protocols to report the contents of the PCRs – and hence the "fingerprint" of the system that has been started – to a remote party.

When a client wants to join a TVD, the TVD master first verifies the integrity of the security kernel of the client, based on security features of the trusted hardware component. For this, an interactive protocol between the security kernel, the trusted hardware, and the TVD master is executed to ensure that only clients that conform to the TVD policy are allowed into the TVD (for details see [20, 5]). After that, a TVD proxy is started on the client, and the TVD master distributes the TVD policy and necessary cryptographic keys to the TVD proxy. The proxy configures the security kernel (e.g., a virtual network for VMs that join the TVD is created), and controls what VMs may join the TVD.

In the German eHC system, man-in-the-middle attacks between client platforms and the healthcare telematics boundary are possible because of missing identification and authentication of the corresponding devices, i.e., the connector box and the client platform [26]. In contrast to this, our proposal allows for mutual device authentication based on security hardware modules attached to the devices, such as the Trusted Platform Module (TPM) [27].

For the communication between the connector and the client platform a secure channel is proposed, but not enforced in the German eHC system. Encryption of the communication is optional, and an authentication of the devices is missing [25, 26]. In contrast, with our proposed architecture and the usage of TVD technology, all client platforms and software components running on them are authenticated by means of attestation functionality using trusted computing technology [27]. Only successfully authenticated components and platforms will be able to establish a trusted channel to the central e-health infrastructure in order to access data of the corresponding privacy domain.

TVD Implementations.

To implement a TVD, a security kernel with support for virtualization and Trusted Computing is needed. We have implemented TVDs as research prototypes (see, e.g., [5]), and a Common Criteria protection profile[6] for a security kernel with support for Trusted Computing functionality has been certified [19]. Operating systems evaluated and certified according to this protection profile would constitute an appropriate basis for industry-grade TVDs.

We realized TVDs based on different virtualization technologies, using results and experiences from our research and development projects EMSCB[7] and OpenTC[8].

[5] An adversary could, e.g., boot a different operating system and try to attack the system.

[6] The Common Criteria are an international standard that aims at permitting comparability between the results of independent security evaluations [8]. A protection profile is a template for the evaluation of a concrete product: it specifies implementation-independent security requirements for a class of products.

[7] See http://www.emscb.com

[8] See http://www.opentc.net

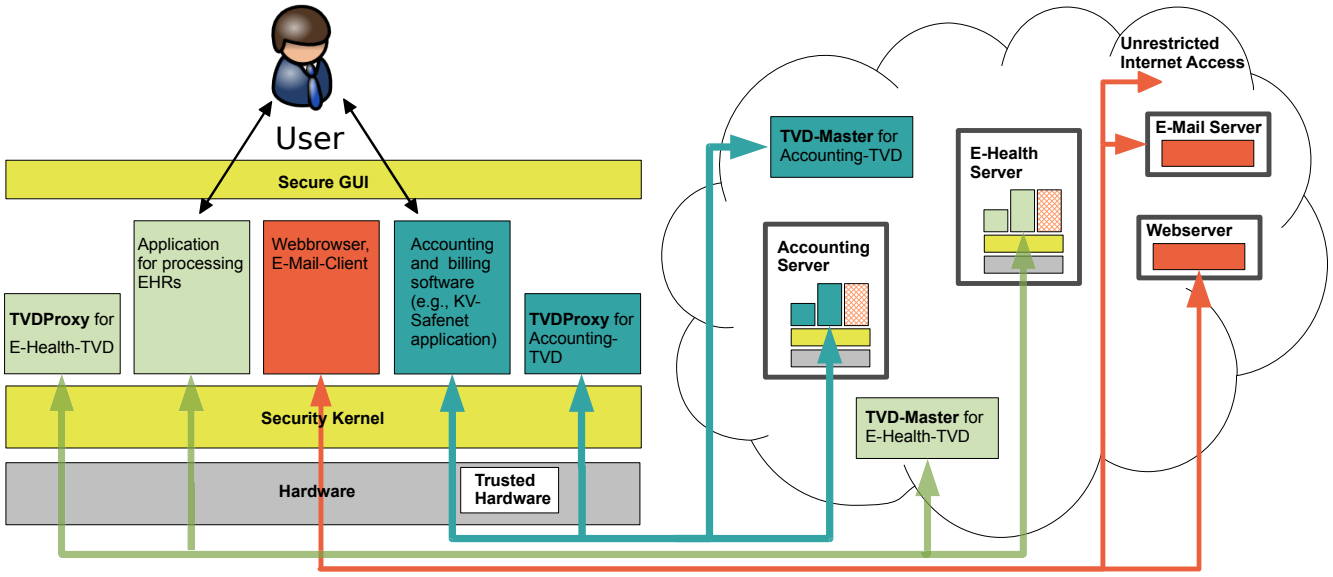

Figure 4: TVDs with client platform, servers, and TVD masters.

We implemented Trusted Computing support based on the TPM, due to its widespread availability. We use the authenticated boot process and attestation functionality of the TPM for the TVD master to verify the client platform integrity, and for the protection of cryptographic keys.

We developed (from previous project results) two interoperable implementations based on the XEN hypervisor [1] and the L4 microkernel [18], with various Linux and Windows versions as guest operating systems. Currently, we are working on a TVD implementation using OpenSolaris, where Solaris zones can be used as guest systems[9]. Sirrix security technologies[10] is offering a commercial product line called Turaya, which supports TVDs [6].

In summary, TVDs have been shown to be practical in a number of different contexts. Although some implementations are research prototypes rather than production-ready systems, various implementations based on different security kernels exist, supporting all kinds of operating systems.

Protection of External Storage.

The secure incorporation and usage of external storage in TVDs, e.g., USB disks, pen drives or cloud storage such as Amazon S3, increases the flexibility of users in their work flows, but requires a careful design of the overall security architecture. Mobile storage devices are regularly employed to store copies of documents that the user (e.g., a doctor) may take home or to another office, or data to be processed on other systems. In particular, USB disks are frequently used *offline*, i.e., plugged to any platform while it is not connected to the domain network (e.g., a laptop in the train or on an airplane). Cloud storage may provide a very convenient and relatively inexpensive way to store backups: While local storage devices or file servers provide quick access to data and are available independently of a working Internet connection, regular backups can be stored conveniently using external cloud services that provide the user with potentially unlimited storage capacity. Only the actual amount of storage that is currently used has to be payed, and the available storage increases as needed. Similarly, file servers that are not part of a TVD can be used as external storage within a TVD.

Storage devices and services should be considered as "passive" components that do not provide security properties. Thus the enforcement of security policies relies entirely on the computer to which the storage is connected. We may assume the policy is correctly enforced as long as storage is used within the TVD boundaries. This assumption is in general no longer true if external storage is used outside its domain, e.g., when it is connected to an outsider computer.

We extended the TVD model with the benefits of using mobile storage devices, allowing the transparent binding of devices to a certain TVD so that only platforms of the same TVD can access the stored data. In the same way, other external storage – such as storage provided by Cloud Computing – can be incorporated into TVDs. From the point of view of the TVD architecture, the storage service provides a container for data, just like a mobile storage device.

Deploying external storage within the TVD requires some refinement to the model due to the following concerns:

- *Storage container identification.* External storage can be moved to one workstation or to another, without any control by the TVD infrastructure. Hence, whenever external storage is connected to a platform, the system should be able to distinguish the device and the domain this device belongs to.

- *Dynamic storage management.* Unlike hard disks that are built into a computer, external storage may unpredictably appear and disappear within the domain, because users may plug in or unplug devices arbitrarily, and network connections to an external server or storage provider may be established or interrupted at

[9] See http://www.trust.rub.de/projects/tvd-solaris
[10] See http://www.sirrix.com

any time. This requires the introduction of a storage management infrastructure in order to handle, e.g., creation and distribution of encryption keys.

To extend TVDs with such a solution for external storage, we enhanced the TVD master with the necessary key management. Moreover, we added a storage device manager to the client's security kernel that identifies the appropriate TVD for storage devices and retrieves keys from the TVD master (if they are not already available in a local cache). The device manager also creates a virtual storage device for the VM of the correct TVD (for details, see [7]).

Security Considerations Concerning External Storage.

Introducing external storage into a system can lead to new security risks. It might be easier for attackers to gain physical possession of external devices than to obtain an internal hard disk. However, due to the transparent encryption of all external storage by the TVD infrastructure, outside attackers who are not part of the TVD cannot access the data.

Moreover, viruses or Trojan horse programs could be stored on USB sticks that are plugged into a system. On commodity operating systems such as Windows, this is a serious threat because programs from USB storage will be automatically executed when the device is attached, and even when the automatic execution of programs from USB storage is disabled, users could manually start malware-infected programs from USB sticks.

With the TVD architecture, we can prevent malware from entering the TVD via USB sticks, because only data from a storage container belonging to the same TVD as a given compartment will be connected to that compartment by the security kernel. Encryption and decryption happens transparently for the compartment of the TVD, and the data cannot be accessed from outside the TVD. For the security kernel, external storage such as a USB disk is just a passive storage device. No programs will be executed from it automatically, neither in the security kernel, nor in any TVD.

Malware that is stored on USB sticks from *within* the TVD cannot be prevented from being read or executed in another compartment of the same TVD (which might be on an different computer system). The TVD infrastructure itself only isolates and protects the TVD from adversaries from the outside, not from malicious software that is already part of the TVD. However, the TVD infrastructure should help to ensure that only secure systems can become members of the TVD by mechanisms such as the integrity verification of all platforms before joining.

5. OPEN RESEARCH CHALLENGES

There are a number of issues with electronic health data that need to be taken into account by systems for EHRs, which are not completely solved by current proposals:

- *Absence of the patient:* The patient is not necessarily present when the EHR needs to be accessed. In this case, using an eHC with a PIN does not work.

  For this, various example scenarios exist: Often, the data is entered into the system only after the patient left the doctor. Moreover, the patient is not present at the doctor's office during preparation of a visit by the doctor at the patient's home. Furthermore, a patient might not be present in person, but is represented by a relative or friend, or a patient consults a doctor remotely, e.g., by phone.

- *Inability of the patient to authenticate:* The patient might be unable (physically or mentally) to remember and enter a PIN.

  Examples scenarios include elderly patients and handicapped people who cannot authenticate by entering a PIN. In emergencies, e.g., in case the patient is unconscious, the patient must be represented by someone else. Moreover, in particular people who only need to authenticate infrequently, tend to forget their PINs.

- *Confidentiality of existence:* The mere existence of an EHR for a given person could already imply that this person received medical treatment, and thus must be kept confidential to avoid violating privacy laws.

- *Client anonymity:* Client anonymity is often not considered at all, but in the context of healthcare, a patient's privacy might be violated by tracking of users or client systems in some scenarios. For instance, if a patient buys medicine in a pharmacy using an electronic prescription, the pharmacist should not be able to trace or identify the patient.

- *Non-repudiation of emergency access:* In case of emergency, health professionals might need to access data urgently in situations, where the patient is unable to authorize this. In such cases, access should be possible, but is important for legal reasons that the person accessing the data can be identified and held responsible. Moreover, this person should not be able to deny the fact that he/she accessed the data.

These issues are not adequately addressed by most current e-health systems, and hence are important research challenges to address. Note that solutions to these problems are orthogonal to the network and platform security issues addressed by our work on privacy domains for e-health systems. We anticipate that future solutions can readily be integrated into TVD-based privacy domains.

6. CONCLUSION

In this paper, we considered security and privacy issues in modern distributed e-health systems, as well as existing proposals and solutions. We addressed an often neglected problem: the security of client platforms. We showed how privacy domains can be used to extend the protection of e-health systems from (existing) network security solutions to a more comprehensive infrastructure, including the client.

As future work, we will address some remaining open research challenges, outlined in Section 5, in our ongoing projects. In particular, we aim to integrate e-health card systems or alternatives into privacy domains and address usability problems in this area.

7. ACKNOWLEDGMENTS

This work was partially funded by the federal state North Rhine-Westphalia under the project RUBTrust/MediTrust.

8. REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. L. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *19th ACM Symposium on Operating Systems Principles (SOSP'03)*, pages 164–177. ACM Press, 2003.

[2] S. Berger, R. Cáceres, D. E. Pendarakis, R. Sailer, E. Valdez, R. Perez, W. Schildhauer, and D. Srinivasan. TVDc: Managing security in the trusted virtual datacenter. *Operating Systems Review*, 42(1):40–47, 2008.

[3] A. Bussani, J. L. Griffin, B. Jansen, K. Julisch, G. Karjoth, H. Maruyama, M. Nakamura, R. Perez, M. Schunter, A. Tanner, L. V. Doorn, E. A. V. Herreweghen, M. Waidner, and S. Yoshihama. Trusted Virtual Domains: Secure foundations for business and IT services. Technical Report RC23792, IBM Research, 2005.

[4] S. Cabuk, C. I. Dalton, K. Eriksson, D. Kuhlmann, H. V. Ramasamy, G. Ramunno, A.-R. Sadeghi, M. Schunter, and C. Stüble. Towards automated security policy enforcement in multi-tenant virtual data centers. *Journal of Computer Security*, 18(1):89–121, 2010.

[5] L. Catuogno, A. Dmitrienko, K. Eriksson, D. Kuhlmann, G. Ramunno, A.-R. Sadeghi, S. Schulz, M. Schunter, M. Winandy, and J. Zhan. Trusted Virtual Domains – design, implementation and lessons learned. In *International Conference on Trusted Systems 2009 (INTRUST'09)*. Springer Verlag, 2009.

[6] L. Catuogno, H. Löhr, M. Manulis, A.-R. Sadeghi, C. Stüble, and M. Winandy. Trusted Virtual Domains: Color your network. *Datenschutz und Datensicherheit (DuD)*, 5, 2010.

[7] L. Catuogno, H. Löhr, M. Manulis, A.-R. Sadeghi, and M. Winandy. Transparent mobile storage protection in trusted virtual domains. In *23rd Large Installation System Administration Conference (LISA'09)*. USENIX Association, 2009.

[8] Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation, Version 3.1*, July 2009. http://www.commoncriteriaportal.org/thecc.html.

[9] Y. Gasmi, A.-R. Sadeghi, P. Stewin, M. Unger, and N. Asokan. Beyond secure channels. In *2nd ACM Workshop on Scalable Trusted Computing (STC'07)*, pages 30–40. ACM Press, 2007.

[10] Gematik. Einführung der Gesundheitskarte - Gesamtarchitektur, Version 1.7.0. http://www.gematik.de/upload/GA_ZentraleDienste_5171.zip, August 2009.

[11] Gematik. Einführung der Gesundheitskarte - Netzwerkspezifikation, Version 2.0.0. http://www.gematik.de/upload/GA_ZentraleDienste_5171.zip, August 2009.

[12] Gematik - Gesellschaft für Telematikanwendungen der Gesundheitskarte. http://www.gematik.de.

[13] J. L. Griffin, T. Jaeger, R. Perez, R. Sailer, L. van Doorn, and R. Cáceres. Trusted Virtual Domains: Toward secure distributed services. In *Proceedings of the 1st IEEE Workshop on Hot Topics in System Dependability (HotDep'05)*, June 2005.

[14] Health Level Seven International (HL7). http://www.hl7.org.

[15] C.-Y. Hsu, Y.-C. Chen, R.-C. Luo, H.-H. Rau, C.-T. Fan, B.-S. Hsiao, and H.-W. Chiu. A resource-sharing platform for trading biomedical intellectual property. *IT Professional*, 12:42–49, 2010.

[16] International Organization for Standardization (ISO). Technical Committee 215, Health Informatics. http://www.iso.org/iso/iso_technical_committee?commid=54960.

[17] Kassenärztliche Bundesvereinigung. KV-SafeNet homepage. http://www.kbv.de/12705.html.

[18] J. Liedtke. On micro-kernel construction. In *Fifteenth ACM Symposium on Operating System Principles (SOSP'95)*, pages 237–250. ACM Press, 1995.

[19] H. Löhr, A.-R. Sadeghi, C. Stüble, M. Weber, and M. Winandy. Modeling trusted computing support in a protection profile for high assurance security kernels. In *Trusted Computing, 2nd International Conference, Trust 2009*, volume 5471 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 2009.

[20] H. Löhr, A. R. Sadeghi, C. Vishik, and M. Winandy. Trusted privacy domains – challenges for trusted computing in privacy-protecting information sharing. In *Information Security Practice and Experience, 5th International Conference, (ISPEC'09)*, volume 5451 of *Lecture Notes in Computer Science*, pages 396–407. Springer, 2009.

[21] H. Löhr, A.-R. Sadeghi, and M. Winandy. Patterns for secure boot and secure storage in computer systems. In *4th International Workshop of Secure System Methodologies Using Patterns (SPattern 2010), In Proc. of Fifth International Conference on Availability, Reliability and Security (ARES'10)*, pages 569–573. IEEE Computer Society, 2010.

[22] H.-H. Rau, C.-Y. Hsu, Y.-L. Lee, W. Chen, and W.-S. Jian. Developing electronic health records in Taiwan. *IT Professional*, 12:17–25, 2010.

[23] T. Schabetsberger, E. Ammenwerth, S. Andreatta, G. Gratl, R. Haux, G. Lechleitner, K. Schindelwig, C. Stark, R. Vogl, I. Wilhelmy, and F. Wozak. From a paper-based transmission of discharge summaries to electronic communication in health care regions. *International Journal of Medical Informatics*, 75:209–215, 2006.

[24] D. Sofsian. An introduction to medical billing. http://www.e-healtharticles.com/Detailed/1449.html, April 2006.

[25] A. Sunyaev, A. Kaletsch, C. Mauro, and H. Krcmar. Security analysis of the german electronic health card's peripheral parts. In *ICEIS 2009 - Proceedings of the 11th International Conference on Enterprise Information Systems, Volume ISAS, Milan, Italy, May 6-10, 2009*, pages 19–26, 2009.

[26] A. Sunyaev, J. M. Leimeister, and H. Krcmar. Open security issues in german healthcare telematics. In *HEALTHINF 2010 - Proceedings of the 3rd International Conference on Health Informatics*, pages 187–194. INSTICC, 2010.

[27] Trusted Computing Group. *TPM Main Specification, Version 1.2 rev. 103*, July 2007. https://www.trustedcomputinggroup.org.