Emergency Access to Protected Health Records

Julien KÜNZI a,b, Paul KOSTER a, Milan PETKOVIC a,1
a Philips Research Europe, Philips Electronics, Eindhoven, The Netherlands
b School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland

Abstract. Digital Rights Management (DRM) schemes are receiving increased attention in the healthcare domain for the protection of sensitive health records as they offer security against insider attacks and advance protection features such as usage control. However, to be accepted by health care providers, a DRM solution has to fulfill specific healthcare requirements including emergency access. In this paper, we propose such DRM solution that can be deployed in highly distributed environments of electronic or personal health record infrastructures.

Keywords. security, EPR-CPR-EMR, telehealth

1. Introduction

Personal and Electronic Health Records (PHRs/EHRs) imply open architectures and interconnected environments. Sensitive patient records no longer reside on mainframes physically isolated within one single healthcare provider, where physical security measures can be taken to defend the data. Patient files are rather kept in distributed environments, where data can be accessed remotely by family doctors, specialists, pharmacies and even non-medical care providers. For example, the national EHR infrastructure (NICTIZ) in the Netherlands employs a decentralized system, where data created under the responsibility of a healthcare provider always resides on the local storage system. Data can be accessed remotely through a central hub, which re-directs queries for specific patients’ documents to the systems of appropriate care providers.

In most countries, EHR systems face strict security regulations (such as EU Directive 95/46 or HIPAA in the US). These laws also protect patient privacy by limiting the use and access to medical records. Furthermore, during the data life-cycle different stakeholders may also specify their privacy policies. For example, a doctor may forbid access to his specific notes or a patient may want a specific care provider to be denied access to his files. The combined rights and restrictions concerning a piece of health information constitute its access policies that define who is allowed to do what.

Traditional data protection models, which lock sensitive data in a server, use a static access control mechanisms and rely on secure connections from the database server to all clients. These models cannot efficiently deal with the requirements of an open interconnected network. The measures they take are not sufficient in a system where data is distributed and accessed by many actors, who are not always part of

1 Corresponding Author: Prof. Dr. Milan Petkovic, Philips Research, Koninklijke Philips Electronics N.V., High Tech Campus 34, Eindhoven 5656AE, The Netherlands; E-mail: milan.petkovic@philips.com.
medical staff (e.g., system administrators, wellness centers, health insurances, etc.). Rather than relying on a central access control policy and secure communication links, Digital Rights Management (DRM) schemes employ a data-centric protection approach: sensitive data is encrypted (using individual keys) and accompanied with a license, specifying the access and usage control policy of the particular data item. Security is enforced at the end-points of the communication by the use of compliant viewers, which allow data access/use only in case the license conditions are fulfilled. Although this approach creates some challenges with respect to data availability (as data is stored encrypted) modern applications of cryptography in the financial and military domain have shown high reliability of these technologies. An overview of the other challenges of DRM in the healthcare domain can be found in [1].

The use of DRM in the healthcare domain faces a dilemma: while medical data needs to be protected, preferably using cryptographic techniques, it must be instantaneously available in emergency situations. Availability of data is more important than confidentiality and crucial data must be made available in an emergency situation to any clinicians (event to the ones that normally do not have access to it), irrespective of the employed data protection model. Modern protection systems for health data must therefore offer special interfaces for emergency access. Due to the difficulty of modeling an “emergency situation” in access control policies, the use of the emergency access mechanism is usually not bound to specific formal requirements. This opens the possibility of misuse, as the emergency mechanism can potentially be used to bypass the entire protection system. To deter illicit access a logging facility must be in place.

In this paper, we present an example of a data-centric protection model applied for protection of health records. In such a model, where sensitive data is encrypted and access is granted according to licenses, we implement emergency access by providing a special emergency license. This license contains an emergency key that can be used to access encrypted data, even if a party does not have explicit access permissions. The scheme is deployed in conjunction with a secure audit logging facility. Since emergency accesses will occur rather frequently the risk that an emergency key gets compromised is high. Thus, the emergency mechanism is designed in such a way that the effects of a compromised emergency key on the entire protection system are limited.

The rest of the paper is organized as follows. In Section 2 we review related work on EHR protection systems that incorporate emergency access mechanisms. In Section 3 we describe a reference architecture for DRM-based EHR systems, which is further enhanced with the proposed emergency access mechanism in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

Nowadays, most hospitals give priority to fulfilling the data availability requirement ensuring that paramedics are allowed access to patient data in an emergency case. Data is often stored in a central database system that employs access control. Emergency access to the data is possible through a “break-the-glass” functionality that allows care providers to exceed their rights and get access to data which they normally would not be able to access. The “break-the-glass” functionality has to be implemented together with non-repudiation mechanisms for its usage. A care provider that accesses data through this mechanism is warned that he is accessing data beyond his normal rights and that this access will be subject to auditing. An audit-log is created and a
notification that the user exceeded his rights is sent to a responsible officer or his superior. Examples of such systems are described in [2–4]. However, most of these approaches assume a centralized system, while, as mentioned above, health data is naturally distributed. In this paper, we are considering distributed PHR/EHR architectures that deploy DRM functionality for sharing health data within a semi-trusted or non-trusted domain, which we extend by implementing the “break-the-glass” functionality and a continuous logging mechanism.

3. System Design

Figure 1 shows the generic structure of a service-based EHR implementation. This reference EHR architecture is used as a basis for different implementations of national EHR systems such as in the Netherlands (NICTIZ) and Canada (InfoWay). Central to these systems is a communication bus that provides access between all EHR system components. Patient’s health and demographic information is distributed over several databases (DB). These databases are run either by independent organizations or by different participating health care providers (like a GP’s office or hospital).

![Diagram](image)

Figure 1. Generic EHR system infrastructure with DRM components (gray boxes)

This architecture has been extended to support DRM functionality. An example of a possible deployment of DRM-components (gray boxes) is depicted in Figure 1. Sensitive data is kept encrypted in databases. Each document is encrypted by an individual document key $K$ using a symmetric encryption scheme (e.g., AES). These keys are managed by the license server, which extends the authorization component of the standard architecture. When a health care provider wants to access data $ID_D$, he creates a request to a license server. The license server constructs a usage license for the health care provider based on access control policies and request context. The usage license contains an identifier of the health care provider, the identifier $ID_D$ of the document, usage rights that are directly determined by the security and privacy policies (a policy could, for example, limit the possibilities of viewing, printing or forwarding the data), and the document key $K$ that allows the client application to decrypt the data. The key $K$ is contained in the license in encrypted form. We assume that all parties in the EHR environment and all health care providers (i.e., their compliant clients) have certified public keys; thus, $K$ can be encrypted (e.g., using RSA) by using the public key of the entity who is granted access. A health care provider is only able to access DRM-protected health data through compliant clients, which enforce the obtained license. A compliant
viewer can be a separate application (e.g., Adobe Reader) or a plug-in for a browser or the existing data management system the healthcare provider is using.

Finally, the publishing server manages the process of adding new documents to the system. In particular, it is responsible for generating document keys, distributing them to the license server(s), encrypting health data, and managing emergency licenses as described in the next section.

4. Emergency Access

For the presented DRM solution to be accepted in healthcare, it is imperative to include emergency access support. In a DRM-enabled EHR repository, all health records are encrypted with individual document keys and a license server only issues licenses to requesting users if they have appropriate access rights [5]. An emergency situation represents an exception to the normal operation of the system [2–4]: in our case a care provider in an emergency situation should be able to decrypt any data, even if he has no ordinary access rights. Legitimateness of access must be proved later using access logs. Note that the emergency mechanism must always be available and cannot depend on the availability of a license server (as this server may be down or unreachable).

We thus extend the afore-defined DRM architecture and introduce special emergency licenses, which travel alongside the data. Key additional feature of an emergency license is the encryption of document key $K$ by an emergency key $EKID_D$ that is attached to the corresponding document. To access these licenses, cooperation of additional parties is required to issue emergency keys and log emergency events.

The proposed DRM emergency access components are depicted in Figure 2. The architecture contains three new components: the key/revocation center (KRC), log service (LS) and a number of emergency agents (EA), which are responsible for making emergency keys available to compliant clients. The role of the KRC and EAs are to assist users in accessing the emergency license and to reliably log such events. The key/revocation center (KRC) is a trusted third party that possesses the emergency keys. The LS is responsible for central logging of emergency accesses and verifying the legitimateness of emergency accesses. The KRC receives incoming requests for document ID$_D$ from EAs and returns the emergency key $EKID_D$, which corresponds to that document and is used to decrypt the document key $K$. Logs must be forwarded by EAs to the LSs of the data-providing and the data-requesting entities such that all participants are legally protected against data misuse.

Emergency Agents (EA) are software agents that process emergency access requests of clients. EAs can, for example, be part of DRM clients in hospitals or GP offices and may travel wherever the DRM system is deployed. They may also be integrated as a separate service within organizations (e.g., hospitals) and then be accessed by compliant clients whenever necessary. This last option will enhance data availability within organizations, still preventing abusive data redundancy, as it enables caching of emergency keys. Emergency Agents are responsible for issuing emergency keys and consequently giving access to emergency care providers. EAs are required to relay emergency requests from clients to the KRC and LS and return the keys. Next to that, Emergency Agents are responsible for local logging of emergency requests and forwarding these logs to the LS using a distributed forward-secure logging facility. Thus, each emergency access is logged by an EA and consequently by the LS. The big advantage is that this allows offline health data usage and thereby improves availability.
If a care provider wants to access document $ID_D$ in an emergency situation, his software client contacts an available EA and submits an emergency request for document $ID_D$. The EA first logs the request locally; the log contains the identity of the submitter, the requested document $ID_D$ and a timestamp. Furthermore, the EA verifies the identity of the submitter, e.g., by running a cryptographic authentication protocol, and logs the result of the verification process. This log is distributed to the LS. Finally, if the identity of the submitter is positively verified, in online scenarios (all involved components are available) the EA passes the request on to the KRC to obtain the corresponding emergency key. The resulting key is sent over a secure and authenticated channel. Finally, the party who requested emergency access extracts the encrypted document key $K$ contained in the emergency license that is accompanied with document $ID_D$ and invokes the decryption algorithm. Subsequently, he can decrypt the document. In offline scenarios (the KRC is not available), the EA itself prefetches and securely stores appropriate emergency keys which are then always available regardless of Internet connectivity. Therefore, the emergency access can take place locally at the client application (that contains an EA) and without being able to connect to the KRC.

5. Conclusion

In this paper, we proposed a novel emergency access mechanism which can be incorporated into EHR systems that employ DRM-like protection of sensitive data. Due to the use of proposed emergency infrastructure the scheme mitigates the problem of emergency key distribution and can thus be deployed in highly distributed systems. The design gives priority to availability making sure that the additional security supports offline operation and does not increase dependability on online services. The feasibility of the proposed system is verified by developing a demonstrator, while testing, acceptance and validation in healthcare environment is still to be done.

References