A SECURE PERSONAL HEALTH RECORD REPOSITORY

Tiago Pedrosa[1,3], Rui Pedro Lopes[1,3], João C. Santos[2,3], Carlos Costa[3] and José Luís Oliveira[3]
[1]*Department of Informatics and Communication, Polytechnic Institute of Bragança, Bragança, Portugal*
[2]*Department of Electrotechnics Engineering, Coimbra Institute of Engineering, Coimbra, Portugal*
[3]*IEETA, University of Aveiro , Aveiro, Portugal*
{*pedrosa, rlopes*}*@ipb.pt, jcandido@isec.pt,* {*carlos.costa, jlo*}*@ua.pt*

Keywords: EHR, PHR, Security, Repositories, Indexing

Abstract: Due to strict regulatory, ethic and legal issues, Electronic Health Record (EHR) systems have been mainly deployed in federated health care scenarios. This situation have been hindering the wide adoption of EHRs, contributing to delaying the establishment of a competitive market where contributions from different providers could take full advantage of information exchange and regular practitioners' collaboration. Moreover, with the increasing awareness of medical subjects, patients are demanding more control over their own personal data - Personal Health Record (PHR). This paper presents a secure PHR repository which access is controlled through the joint use of a Virtual Health Card Service (VHCS) and an access Broker. This solution can be deployed in any public or private storage service since it behaves as a sandbox system which access policy is defined externally. To assure a friendly query-retrieve interaction the whole repository is indexed, and separated clinical events are kept independently to increase the efficiency of cipher and encipher algorithms.

1 INTRODUCTION

Electronic Health Records (EHR) can be defined as digital record that aggregate all data acquired during patient care on the healthcare system. Soon captured the attention of practitioners, policy makers and patients since they are essential to better clinical services integration and to health information sharing. However, EHRs have mainly been deployed in more or less enclosed health care scenarios. Strict regulatory, ethic and legal constrains (Hodge Jr et al., 1999; Shabo, 2006), have been hindering the wide adoption of EHRs, contributing to delaying the establishment of a competitive market where different providers could take full advantage of information exchange and regular practitioners' collaboration. Moreover patients are demanding more control over their own personal data.

To cope with these new user requirements, several Personal Health Record (PHR) solutions have been developed, enabling users to keep record of their medical data. Examples of such system are, for instance, Google Health, Microsoft HealthVault and Dossia (Eysenbach, 2008). These web-based PHRs are mostly based on a central repository and on a set of core features that, in some cases, can be extended by external third-party services. A major difference between EHR and PHR is related to the responsibility for maintaining the information and for specifying the access control policy. In the former model the accountable are the healthcare institutions and their professionals, while in the PHR it is the patient that owns this responsibility. Despite this typical operational model, several PHR can also be automatically populated from systems where the data is generated. However, for this scenario to be possible, it will be necessary that EHR systems generate the adequate reports of clinical events and that could interact directly with those external PHR systems.

The exchange and storage of health information is a major security challenge, mainly its disruption may compromise seriously personal privacy. The idea of having an enterprise with access to all health information of a citizen is unlikely to occur because the risk of information discloser is higher and more disastrous than in a scattered scenario. Moreover, the *Big Brother* scenario also appears whenever centralization is suggested, despite of the guardian of the information is an enterprise or the government, and this vision also slows down the adoption of the different solutions. This paper presents a secure PHR repository where access is controlled through the joint use of a Virtual Health Card Service (VHCS) and an access Broker. Several security features were decou-

pled along both components to assure a double consensus when manipulating the data behind. It can be deployed in any public or private storage service since it acts as a sandbox system which access policy is defined externally. To assure an efficient user interface, the whole repository is indexed and separate clinical events are kept independently to increase the efficiency of cipher-decipher and query-retrieve algorithms. This strategy enables the citizen to safely deposit information on the PHR repository, since not even the repository administrator can discloser it. Under an explicit owner's authorization, the Broker may enable other services to access or upload new data in the PHR.

2 BACKGROUND

EHRs and PHRs can share the record architecture, but they differ in the data custody ownership and the management responsibilities. The PHR can be a self-contained registry, maintained and controlled by the subject of care, based on a specific portable data storage device, some entry in a web service provider or even a component of an Integrated Care EHR (ICEHR). In the EHR case healthcare providers are responsible for its maintenance. Different types of EHR exist, but one that is more promising is the ICEHR, that acts as a repository of all the health information of a patient during is life time (ISO/TC 215, 2005).

For achieving a functional EHR, interoperability between producers and consumers of information is needed. Standardization appears as the solution to enable the communication between different systems. These efforts can be divided in two main areas: the communication standard and the document standard (Sunyaev et al., 2008). The former refers to how systems can communicate with each other and the latter describes how information is stored to ensure a correct interpretation by other systems. The results of researching the available standards has evidenced interoperability barriers.

Diverse documents standards exists, as HL7 specifies the Clinical Document Architecture (CDA) (Dolin et al., 2006), ASTM the Continuity of Care Document (CCD) and the Continuity of Care Record (CCR) (Ferranti et al., 2006) that are constrains over the CDA and the OpenEHR (OpenEHR, 2007) uses archetypes for defining the record structure. Those formats have is the ability to extract a record or a subset of a record in a XML format where the DTD defines the record structure. This provides an easier way to export data from one system to another.

The increased mobility and free market in health care provisioning pushed the information to be scattered through the providers. A solution can be the use of an integrated access mechanism to the disperse information. The integrator has to know the location of the information and how to retrieve it in a secure way. This linkage information can be stored in the integrator database, or to an extend electronic health card to support that service (Ferreira Polónia et al., 2005; Costa et al., 2003).

In a wider concept of mobility it is not feasible that all worldwide patients will have the same type of card. Another open question is that services and users can't make use of the information of the patient if the card is not present. To overcome that difficulties was proposed a Service Oriented Architecture (SOA), making use of a Virtual Health Card Service (VHCS), that mimics the behavior of the physical card (Pedrosa et al., 2009). Combined with a centralized access control mechanism that implements the intent consent policy and with a proxy. The proxy uses the information inside a VHC, namely the credentials of the patient to the repository, the access policy to apply to the requester user, and the URLs to the scattered repositories to retrieve the information and create a unique EHR read-only view (Pedrosa et al., 2010).

The tight regulatory framework that health care services have to comply, limits their will to give access to their systems. Therefore it will limit the use of the previous solution. A solution that appears is centralization of the information in one repository. The EHR healthcare system will create and deposit on the repository the information related to the patient attendance. If all systems this procedure, the information in the repository will enable the creation of the ICEHR. PHRs make use of services for importing information, but as the record management is done by the patient, medical staff tends to question the integrity of the data.

One of the main concerns about PHR is its privacy, as the data inside any healthcare provider allows access to any PHR, it can potentially open the system to large-scale disclosures (Ray and Wimalasiri, 2006).

3 PROPOSED SOLUTION

The proposed solution mimics the process for storing safe deposit boxes inside banks. A procurator is responsible for managing the deposited goods, which, in our case, correspond to the information stored in health records. The citizen will authorize the procurator to deposit and withdraw information from the

safe deposit to answer the needs of third party actors, previously approved by the citizen. An additional level of protection is added, by defining several small safes inside a larger safe. The keys for those small safes will be stored in a different and secure place (the VHCS), so the information will not be visible to the bank or to the procurator. When a requester needs access to the information, he will ask the procurator to provide the data and for that he also provides a safe box that can only be opened by the procurator, through the keys stored in the VHCS. The procurator, after consulting the privileges of the requester, will unlock the safe and retrieve the small safe deposit boxes that contain the required information and temporarily store it in a secure place. The citizen's safes will be opened and, according to the requester privileges, the information will be copied to his requesting safe, which will be then returned to him.

For depositing information, the requester asks the procurator to bring a safe that will be used to put the information in. Then, the procurator will take the safe back, and store it in the citizen's larger safe deposit box inside the bank. This approach means that neither the bank nor the procurator can access the information inside the safe, but allows him to manage it. To get into the details of the architecture, we provide an overview of the actors and services involved. The Virtual Health Card Service (VHCS) represent the secure key deposit; the Broker represents the procurator; the Repository represent the bank; and the Indexer will enable a selective retrieval from the Repository (Figure 1). We use a Service Oriented Architecture that makes use of SSL certificates, signed by a Certificate Authority (CA), to establish confidence between the actors and software components. The record is pop-

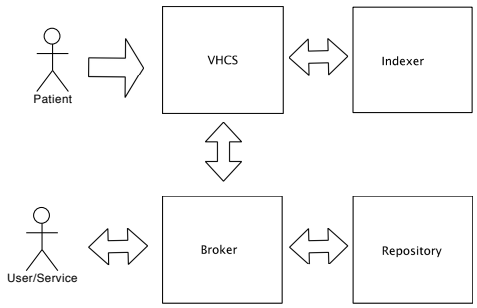

Figure 1: Wide overview of several actors and services

ulated with information resulting from several contributions, associated with the different actors, when they upload the information resultant from medical activity. All the contributions are stored and ciphered individually inside the repository. In the request process these contributions are retrieved from the repository, deciphered, reassembled and ciphered using the requester key. This assures that the information in transit is always encrypted and that the only service that can read the information is the VHCS. In the store procedure, the information generated by a medical exam, for example, is ciphered using the patient's public key and stored in the VHCS. When sending this information to the Repository, it is ciphered with the public key of the receiver.

The Repository is a storage service that enables a client to request or to store bulks of data associated to a specific identifier. The identifier is unique per patient and it is used to associate the information with a specific patient. The only way to associate the stored information to a specific patient is through the VHCS. The Broker is the only service that can contact directly the repository. It acts as a middleman between the users and services that need access to the repository. It has to validate the users and fulfill their requests through the VHCS when needed.

The creation of a Virtual Health Card by the patient is the first step to enable the use of a secure PHR repository. This phase is composed by the patient credentials generation (public/private key pair), the initialization of a repository i.e. configuring the repository end point, and the credentials that will allow the broker to query that Virtual Health Card. The users and services that want to make use of the patient information on the repository or wants to save new information will use the services provided by the Broker (Figure 2). The VHCS components are directly related to patient functions, supported by the following components: the Patient Credential, the Access Control and Repository Management. It also provides functions and interfaces to be used by the Broker.

The Patient Credential manage public and private patient keys. The Cryptographic component is responsible of manipulating those keys in a secure way and it also provides cryptographic functions to manipulate the ciphered block enforcing that the private key never leaves the VHCS. The Access Control component stores the policy defined by the patient to regulate the access of the users and services that want to gain access to the repository. It stores the public key of the user/service and the type of the access he has to the information. The Repository Management component is responsible for managing the Repository store and the credentials required to gain access.

The Reassemble component provides a way to aggregate the requested individual contributions in a single one. During this phase the access control is enforced by the discard of information that the requesting user does not have access.

The Broker interface component provides access to functions that the Broker will need to perform. It

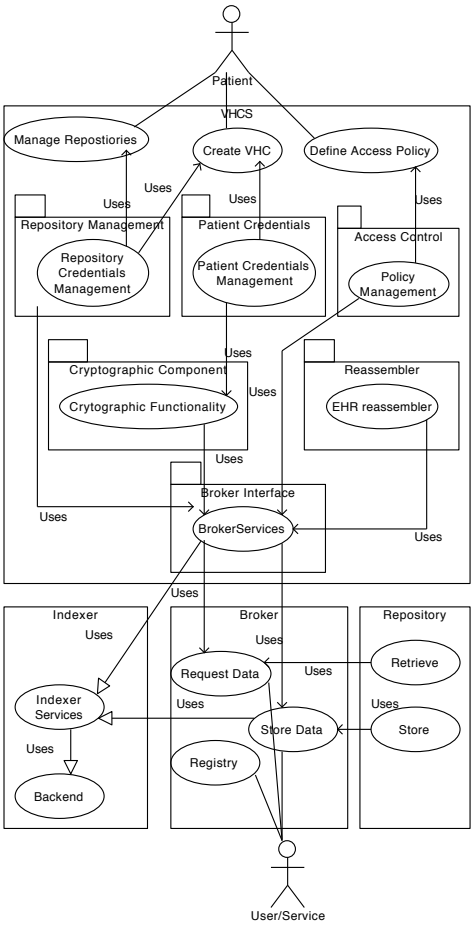

Figure 2: Interaction between users and services components

enables the Broker to check the access policy, to know what Repository to contact and the required credentials. It also enables the Broker to request the public key of a patient and invokes the ciphering/deciphering of data through the Cryptography component using the patient keys.

The Broker components are directly related with actions that other users/services perform on the patient repository. First, the users/services have to create an account on the Broker, which will check for the validity of the CA signature, and stores the public key of the requester. This registry on the Broker is not only used for users/services authentication. A patient can also use this to retrieve the public key of users or services that may want to use the VHCS access control component. The Broker makes use of the VHCS capabilities through the broker interface, and together they enable storing and retrieving of information to and from the Repository. The Broker manipulates the information using a closed envelope concept as the information is always ciphered to the receiver, to the patient or to user/service that request information.

Figure 3 explains how an external service or user can store information on the repository. When a

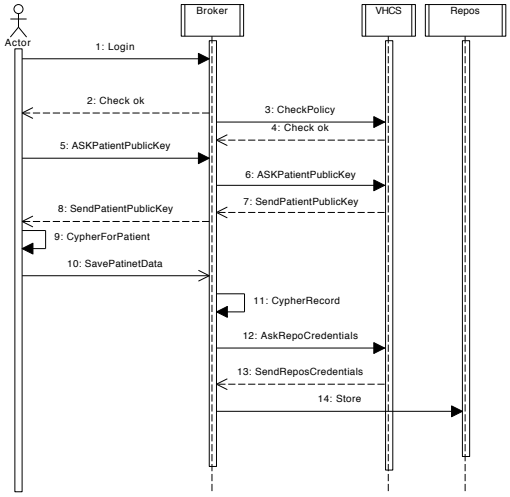

Figure 3: Storing information on repository

user/service wants to save new information of a patient on the Repository it contacts the Broker. First it challenges the login and checks the authorization policy on the VHCS. After success, the actor will ask the public key of the patient. To be able to answer, the Broker requests the patient public key to the VHCS and forwards it to the user/service. The later ciphers the information with the given public key and sends the result back to the Broker. The Broker ciphers again using its private key and requests the patient repository information to the VHCS. Finally, it uses this information to store the information on the Repository.

The first ciphering procedure ensures that the information sent to the Broker can only by read by using the patient's private key, stored on the VHCS. The second ciphering forces the Broker to interact with the information stored on the inside the repository. The double ciphering obliges the Broker and VHCS to cooperate in order to manipulate the data.

Sometimes, the requester only needs access to a subset of information, relieving the burden and the overload of processing and sending the full data set. To optimize this process, an indexing service is provided, enabling some search capabilities over the patient repository. The proposed architecture contemplates an independent indexing service that stores searchable information. This Indexer accepts queries and returns PHR repository entries that fulfill the request, avoiding unnecessary manipulation of data pieces.

Every time a new piece of patient clinical data is archived, the Indexer securely stores a set of associ-

ated meta-information, including the event type, some coded clinical details, creation date, producer information and repository data location. Every patient has its own set of index files that are ciphered with his credentials. When creating new data in the Repository, the actor must create indexing information and cipher it with patient public key. Both information blocks, patient data and index metadata, are sent to the Broker. Here, the index is delivered to Indexer Service via VHCS and the patient record is stored in the Repository.

The Index service is only accessible through VHCS, which controls all access to searching operations. When an actor needs to search information, it requests the patient public key to encrypt the query that is sent to Broker (Figure 4). This provider will check the actor access policies before forwarding the request to the VHCS. Here, the query is decrypted and injected in the Index engine. The query results, i.e. the data repository references, are returned to the Broker via VHCS. The Broker then gets the patient repository information and uses it to retrieve the contributions from the Repository. Next it will decipher it using Broker's private key and send the result to the VHCS along with the public key of the requester. The VHCS deciphers the received information using the patient private key, it reassembles the various contributions and ciphers it using the public key of the requester. In the end, the result will be sent back to the Broker, which forwards the data to the requester. The information detained by Indexer module is only

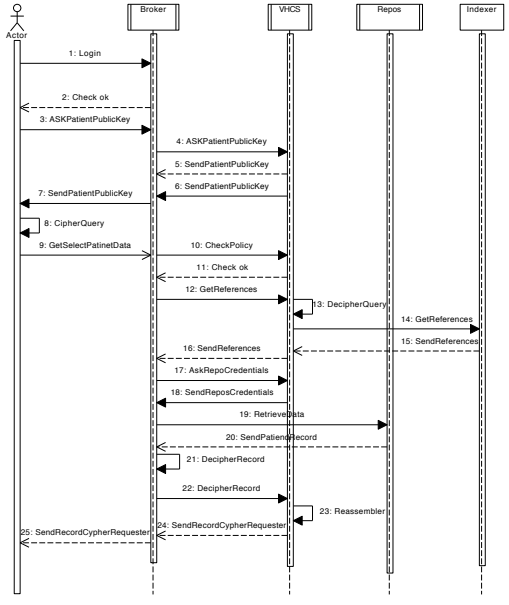

Figure 4: Selective Retrieving from repository

readable by VHCS because it is ciphered with the patient public key. Moreover, the information stored in the Repository and Indexer can only be related by a third secure entity – the VHCS. This approach enables search operations over a ciphered repository, without disclosing patient information during the process. The Broker cannot use actor queries information or Indexer results to extrapolate what kind of data exists inside patient Repository.

This approach grants that the information is always transmitted in a closed envelope concept and that the only actor that can read the information is the requester. Moreover, the Broker and the VHCS have to work together to gain access to the information on the repository.

Although the record can follow virtually any data format, for our proposal, we have chosen XML. Since the record format is transparent in terms of ciphering and storage, the producers and consumers only have to agree on a common format for enabling collaboration in the creation of the PHR. If not, they must at least support interchangeable schemas to enable a common understandable format. The only component that has to be defined according to a specific format is the re-assembler in the VHCS.

The use of XML allows saving the record structure as well the data. The structure can provide information about the type of contribution that exists on the record, even when the data is ciphered. It is true that the data itself could be protected, but a specific structure for saving a type of a lab test result or medical procedure can reveal, the kind of test or procedure the patient made. This would allow guessing patient pathologies and diseases.

4 IMPLEMENTATION

The prototype is being developed in Java EE 6), over Glassfish application server. All components are deployed as Enterprise Java Beans. The interfaces between the components will be implemented via web-services over HTTPS. The client authentication is configured and requested in the application server, being transparent to the developer. The OpenSSL was chosen as public key cryptographic framework and as digital credential management (users and services) platform it was used the OpenCA bundle. A Crypto API (Legion of the Bouncy Castle, 2010) is used to manipulate certificates and for cryptographic functions.

The Repository Service is a file system oriented storage structure, using a unique subdirectory per patient, each with a random identification number. Each patient directory stores the ciphered clinical records

submitted to Repository Service by the Broker. Each structured record file receives a sequence number that unequivocally relates it to the patient index information stored in Index system.

To test this prototype, a Google Health account is being used to create a test record. Using the Data Exchange tool (Mount Tabor, 2010) the record is exported to a file. That file is uploaded to our Repository via our Broker to test the solution. The inverse process will be also tested.

5 CONCLUSIONS

Personal Health Records have recently appeared as a solution that allows patients to keep track of their own clinical history. With the increasing availability of medical information in the Internet, citizens now have access to a tremendous amount of data sources, which provide information such as diseases symptoms, diagnostics, treatments, drugs, physicians, and many others. This situation is leading to better informed citizens, but with much more complex requirements – health information gathering and privacy assurance are two critical examples.

This paper presents a secure PHR repository that combines the notion of a safe deposit with the ability to securely share clinical data. This particular feature enables the integration with external services upon an explicit authorization by the patient. In this way, much of the data that is uploaded in the repository can come directly from EHR systems, which alleviates the user from the burden of record updating. The aggregation of all these contributions enables a comprehensive overview of the patient medical status and relevant historic information. Moreover, patient's personal registries can also be kept in this system.

To assure security requirements, the information in transit is always ciphered. On storage operations, the system uses the patient public key and on retrieval it applies the public key of the requester. In the store procedure a second entity, the Broker, will also cipher the data before storing it in the repository. The only component that can read the sandbox record is a virtual card component (VHCS), but without access to its content. Despite holding the repository credentials and the private key of the patient to decipher the data, the VHCS would need that the Broker also retrieve the data from repository. The proposed solution provides a search capability through an indexing service that maintains links to metadata of all the PHR contributions, allowing a selective retrieve from the repository whenever a query is executed.

REFERENCES

Costa, C., Oliveira, J., Silva, A., et al. (2003). A new concept for an integrated healthcare access model. *The new navigators: from professionals to patients: proceedings of MIE2003*, page 101.

Dolin, R., Alschuler, L., Boyer, S., Beebe, C., Behlen, F., Biron, P., and Shabo Shvo, A. (2006). Hl7 clinical document architecture, release 2. *Journal of the American Medical Informatics Association*, 13(1):30.

Eysenbach, G. (2008). Medicine 2.0: social networking, collaboration, participation, apomediation, and openness. *Journal of Medical Internet Research*, 10(3).

Ferranti, J., Musser, R., Kawamoto, K., and Hammond, W. (2006). The clinical document architecture and the continuity of care record. *British Medical Journal*, 13(3):245.

Ferreira Polónia, D., Costa, C., and Oliveira, J. (2005). Architecture evaluation for the implementation of a regional integrated electronic health record. In Press, I., editor, *Proceedings of MIE2005*. IOS Press.

Hodge Jr, J., Gostin, L., and Jacobson, P. (1999). Legal issues concerning electronic health information: privacy, quality, and liability. *Jama*, 282(15):1466.

ISO/TC 215 (2005). Health informatics - electronic health record - definition, scope, and context - iso/tr 20514:2005(e). Technical report, ISO.

Legion of the Bouncy Castle (2010). The Legion of the Bouncy Castle. `http://www.bouncycastle.org/java.html`, Last Checked: 1 June 2010.

Mount Tabor (2010). Mount tabor and google health. `http://www.mttaboros.com/GHPartners.html`, Last Checked: 1 June 2010.

OpenEHR (2007). Introducing openehr - revision 1.1. Technical report, OpenEHR.

Pedrosa, T., Costa, C., Lopes, R., and Oliveira, J. (2009). Virtual health card system. *Inforum 2009*.

Pedrosa, T., Lopes, R., Santos, J., Costa, C., and Oliveira, J. (2010). Towards an EHR architecture for mobile citizens. In *HealthInf 2010 Proccedings*.

Ray, P. and Wimalasiri, J. (2006). The need for technical solutions for maintaining the privacy of EHR. In *IEEE Engineering in Medicine and Biology Society.*

Shabo, A. (2006). A global socio-economic-medico-legal model for the sustainability of longitudinal electronic health records-part 2. *Methods of information in medicine*, 45(3):240.

Sunyaev, A., Leimeister, J., Schweiger, A., and Krcmar, H. (2008). It-standards and standardization approaches in healthcare. *Encyclopedia of Healthcare Information Systems.*