# Restoring the Patient Control over her Medical History

Nicolas Anciaux, Mehdi Benzine, Luc Bouganim, Kévin Jacquemin, Philippe Pucheral, Shaoyi Yin

# Restoring the Patient Control over her Medical History[1]

Nicolas Anciaux[*], Mehdi Benzine[*,**], Luc Bouganim[*], Kévin Jacquemin[*], Philippe Pucheral[*,**], Shaoyi Yin[*,**]

[(*)] *INRIA Rocquencourt*
*Le Chesnay, France*
*<Fname.Lname>@inria.fr*

[(**)] *PRiSM Laboratory*
*University of Versailles, France*
*<Fname.Lname>@prism.uvsq.fr*

## Abstract

*Paper-based folders have been widely used to coordinate cares in medical-social networks, but they introduce some burning issues (e.g. privacy protection, remote access to the folder). Replacing the paper-based folder system by a traditional Electronic Healthcare Record (EHR) introduces new drawbacks: forcing of the patient consent, unbounded data retention, no security guarantee outside the server domain and no disconnected access to the folder. To solve these problems, this paper proposes an experimental platform which combines an EHR system with medical-social folders embedded in a new hardware portable device. The objectives pursued are (1) to re-establish a natural and powerful way of protecting and sharing highly sensitive information among trusted parties and (2) to build a shared medical-social folder providing the highest degree of availability, whatever the mode of operation (disconnected or not).*

## 1. Introduction

### 1.1. Context of the study

The ageing of population makes the health monitoring of elderly people at home crucial. In this context, sensitive data has to be shared between all participants of medical-social networks (doctors, nurses, social workers, home help and family circle) with different access rights. The data must be available at the patient's bedside for a better monitoring of their health cares. For this purpose, the Yvelines District in France has decided to carry out an experimental project of Shared Medical-Social Folder, called SMSF (DMSP in French) in the following. In the first step, this project targets elderly people from two gerontology networks. At mid-term, it could be extended to other vulnerable people in unstable or handicapped situation.

ALDS (a home care association) has already carried out a "Common Medical Folder" in paper format, which enables professionals and participants from medical-social sectors to write down crucial facts related to the monitoring of elderly people. While the day-to-day use of this paper folder has proved its efficiency, some burning issues are still unresolved:

- *No privacy*: all participants (doctors, nurses, social workers, home help and family circle) can access to the full folder while some patients are facing complex human situations (diagnosis of terminal illness, addictions, financial difficulties, etc).
- *No remote access to the folder*: consequently, the folder is not updated consistently and timely, leading to a lesser accurate monitoring.
- *No connection with computer-based information systems:* this implies later multiple keyboarding of the same data, an error prone and time waste task.

### 1.2. Motivation

During the last decade, several countries launched ambitious Electronic Healthcare Record (EHR) programs with the objective to increase the quality of care while decreasing its cost [4, 5]. The advantage of centralizing the information in EHR systems is manifold: completeness (i.e., to make the information complete and up to date), availability (to make it accessible through the internet 24h-7 days a week), usability (to organize the data and make it easily to be queried and interpreted), consistency (to guarantee integrity constraints and enforce atomicity and isolation of updates), durability (to protect the data against failure) and security (to protect the data against illegal accesses).

[1] This project is partly funded by the Yvelines district council and by ANR (the French National Agency for Research).

1063-7125/08 $25.00 © 2008 IEEE
DOI 10.1109/CBMS.2008.101

On the other hand, studies in different countries [7, 13] shown that patients are reluctant to use EHR systems arguing increasing threats on individual privacy, whatever the security procedures placed at the server. Actually, the patient has the feeling to definitively loose the control over his data for the following reasons:

– *Forcing of the patient consent:* the high number of involved people, the diversity of their roles and the intrinsic complexity of the medical information force the patient to blindly adhere to a predefined access control policy that he does not really master. Complementary to access control, audit trails can help the patient tracking a posteriori who accessed which part of his folder and when. However, the complexity of auditing increases with the size of the trail [3]. With respect to the free expression of the patient consent, EHR systems cannot compete with the archaic paper-based information sharing.

– *Unbounded data retention:* Data retention is a usual law principle attaching a lifetime to the data after which it must be withdrawn from the system [6]. Unfortunately, data retention conflicts with the primary objective of an EHR, which is building a complete medical history of a patient. In addition, [9] highlighted the difficulty to physically destroy data stored in existing DBMSs. This reinforces the patient feeling that his complete history is recorded forever. As a result, the patient may choose not storing some information in his folder (synonym of incompleteness and lower quality of care).

– *No security guarantee outside the server domain:* healthcare data is likely to be extracted from the server and hosted in a client device (e.g., the doctor's device) for use in a disconnected mode, e.g., to provide care at home. Unfortunately, the hosting device is much more spyware, Trojan and virus prone than the server, introducing a severe security breach in the architecture.

– *No disconnected access to the folder:* EHR have been designed with an on-line usage in mind. Thus, the prerequisite for a large category of patients (e.g., elderly, disabled and needy people) to get access to their folder is either to use a terminal at some public place or to own a PC and to pay for an internet connection. Otherwise, a practitioner providing care at home will have beforehand to download on his mobile device the folders of all visited patients, a complex and time consuming task, apart the security breach mentioned above.

The motivation of our work is precisely to address these four issues.

### 1.3. Problem statement

According to the discussion above, the dimensions of the problem tackled by the SMSF project are:

1. *Agreement expression:* to give back the ability to the person to establish a strict, understandable and auditable control on how his sensitive data are shared.
2. *Data retention:* to give back the management to the person for the data retention period, by distinguishing the notions of retention and durability.
3. *Security continuity:* to guarantee the same security level wherever the data is hosted (server or terminal) and whatever the manner it is processed (remote or offline access).
4. *Disconnected access:* to enable disconnected accesses to, and modification of, the data while guaranteeing eventual data consistency.

In this paper, we propose a novel organization of EHR addressing this problem statement. This organization capitalizes on a new hardware device called Secure Portable Token (SPT) in the sequel. Roughly speaking, an SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized) on a USB key form factor [8]. An SPT can host on-board data and run on-board code with proven security properties thanks to its tamper-resistant hardware and a certified operating system. Embedding a database system and a web server in an SPT gives the opportunity to manage (part of) a healthcare folder outside the EHR server with no loss of security. Accessing the on-board folder while being disconnected from the network requires a simple rendering device equipped with a USB port and running a web browser. More, the embedded DBMS can be made self-administered so that the patient keeps a full control over the on-board data.

The paper is organized as follows. Section 2 introduces the principles of the solution and illustrates them through typical scenarios. Section 3 presents the functional architecture. Section 4 sketches important technical challenges related the management of databases embedded in SPT. Section 5 concludes.

## 2. Principles and Scenarios

### 2.1. Basic Principles

We illustrate below how the SPT technology complements the traditional EHR system to cope with the exposed problems. Each user U owns (1) a personal healthcare folder managed by a central server and (2) a personalized SPT. This SPT contains the numeric certificate of U enabling him a strong authentication to the server when he remotely

accesses to his folder. This SPT contains also a (total or partial) replica of U's healthcare folder. Finally, the SPT contains embedded software components (Web server and DBMS) giving offline access to the folder through any terminal hosting a USB port and a web browser.

Since U may exchange data with a health professional P, these professionals have also to be equipped with a personalized SPT. The health professional P's SPT is similar to the one of U on a software and hardware point of view, but its role in the interactions with U's folder is particular. P's SPT contains P's certificate enabling a strong authentication to the server, whether it is the central server or the server embedded in U's SPT and managing U's folder. In both cases, P follows the access control policy fixed by U, which is the same on the central and the embedded server.

If P accesses to U's data and decides to replicate it on his terminal, for example to have offline access to it, the data will be encrypted with a key only known by P's SPT. When P queries the data, the embedded DBMS in his SPT accesses and decrypts it to answer the query. A similar principle allows U to dump encrypted data on the central server in destination of P, who will be able to access to it thanks to the DBMS and the keys embedded in his SPT (assuming the keys have been shared between U's SPT and P's SPT through a so-called secure channel). The difference for U between dumping clear or encrypted data on the central server is as follows. In the first case (clear data), data sharing is controlled by the access control policy that U consented to. In the second case (encrypted data), the access control policy is strengthened by an obligation of physically sharing the encryption keys, and this sharing is managed totally under the control of U.

To organize the sharing, user U can choose different status for his data:

- Regular data: regular data is replicated on the central server and the SPT, protected by the same access control policy. The motivation to replicate regular data in the embedded folder is to assure their offline availability.
- Secret data: secret data is exclusively stored on U's SPT. U keeps the freedom to grant access to his SPT, and thus to the secret data, to the health professional P who is physically in front of him. He is guaranteed that nobody can access to his data without his awareness. On the other hand the durability of secret data is not guaranteed.
- Resilient secret data: it is the secret data replicated on the central server in an encrypted format using encryption keys exclusively known by U's SPT. The server maintains the durability of these data like regular data but U keeps the guarantee that nobody can access to them (i.e., in plain-text) without holding U's SPT. U is just in charge of the durability of the encryption keys (e.g., thanks to a passphrase).
- Confined data: it is the data that U wants to share in an exclusive manner among a reduced circle of trusted persons, with the guarantee that nobody else can access to the data. To do this, U puts encrypted data on the central server via his SPT, and shares these keys with the SPTs of the trusted circle.

Finally, the data replication between central server and embedded server raises the tricky problem of synchronization. When two servers are connected with each other, the synchronization is done traditionally. However, this situation may never occur (e.g. a patient U may never leave home). In this particular case, a synchronization protocol using proxy will take place, in which the SPTs of health professionals in touch with U are used to carry encrypted messages between the central and the embedded server, similarly to a – slow – network.

## 2.2. Sharing scenarios

Let us illustrate the behavior of the system through scenarios involving three participants: an elderly patient named Bob, his family doctor Jim, and a nurse Lucy. Every participant owns an SPT. Several medical examinations are prescribed to Bob who designates a subset of them as confined (the others being considered as regular). The medical lab performing the examination pushes the results on the central server. Results corresponding to confined data are crypto-protected using Bob's public key2 before being pushed.

Lucy frequently visits Bob at home. Bob has no internet connection and leaves home seldom. Thus, Lucy acts as a synchronization means for Bob's folder (as any other person visiting Bob and owning an SPT). Before the visit, Lucy downloads from the central server only the latest updates, either confined or regular, performed in Bob's folder. This includes the recent examination results. During the visit, Lucy's and Bob's SPTs are synchronized. The latest updates from the central server are integrated in Bob's local folder. Conversely, the latest updates

[2] Bob's public key is delivered by a PKI server while Bob's private key is replicated on every SPT belonging to Bob's trusted circle. The management of private keys is under the control of the secure chip and even the SPT holder cannot interfere or tamper it. For the sake of conciseness, we do not detail further the key exchange protocol among SPT. For efficiency, asymmetric encryption is used only to encrypt symmetric keys used to protect the confined data.

performed in Bob's local folder, if any, are loaded on Lucy's SPT. This allows refreshing the central server replica the next time Lucy connects to the server. If Lucy does not belong to Bob's trusted circle, she cannot get access to the carried confined data nor to the unauthorized regular data, because (i) her SPT does not own Bob's private key, and (ii) the access control module embedded in the SPT regulates access to the regular data.

Jim participates in Bob's trusted circle. At his office, he can connect to the central server and view Bob's up-to-date folder, including the results of the recent examinations and possible updates carried back by Lucy. When visiting Bob at home, he can get the same level of information by connecting locally to Bob's SPT.

As a conclusion, any authorized people can connect to the central server or to an SPT local server and retrieve the regular data he is granted access to by the access control policy. No people outside the trusted circle can get access to the confined data. Nobody except the SPT owner can access his own secret data or resilient secret data. These confined and resilient secret data are protected against any form of attacks. Confined and resilient secret data are crypto-protected on the central server and attacks on the SPT are made extremely difficult by the tamper-resistance of the chip and by the fact that the DBMS is self-administered.

# 3. Functional architecture

Figure 1 represents the functional architecture of SMSF which distinguishes infrastructure, data and software aspects. At the software layer, the shaded components are commercial software components (Web server, Web browser, relational DBMS, operating system), while the other components are designed and implemented particularly for the SMSF project.

## 3.1. Infrastructure

The infrastructure part in Figure 1 (from left to right) represents the central server hosting the personal folders, a user P who is connected through a laptop and potentially having downloaded part of the database (to use it in a disconnected mode), and a user U synchronizing his SPT on a PDA thanks to the updates transmitted by P's SPT.

The connection of users to the server is done through a secure TLS (Transport Layer Security) channel which guarantees the security of the communication. A strong mutual authentication occurs thanks to the certificates embedded in each SPT.

The server provides a web interface to consult, create and synchronize data in a personal folder through a web browser. All these actions are done through the secure channel mentioned above.

This infrastructure is based on classical hardware except the SPT.

## 3.2. Data

The data part in Figure 1 shows the different kinds of data and their encryption format in each environment, according to their status. The central server stores administration data such as the users' certificates and their public keys. The server also stores secret keys (k) which are encrypted by using the public keys of the users belonging to the same trusted circle. Then, the central server stores, for each healthcare folder, the Regular Data (protected by the security procedures enforced by the database service provider), the resilient secret data encrypted with one (or more) secret keys $k_U$ known only by U's SPT, the confined data encrypted with one (or more) secret keys k. The user P's terminal can host the data to make them reachable in disconnected mode. These data are encrypted by using one (or more) secret keys $k_p$ known only by P's SPT. Finally, an SPT hosts different kinds of data depending on whether it is used (1) as a user SPT to deal with a personal folder or (2) as a professional SPT to access a user's folder (note that the two roles may be played at the same time). In the first case (a user dealing with his own folder), the SPT hosts administrative data such as the user's certificate, his private key (the counterpart of his public key hosted by the central server), a set of secret keys $k_U$ used to encrypt the resilient secret data on the central server, a set of secret keys k used to encrypt confined data. These administrative data are stored in the secured microcontroller's internal NOR Flash so that they are protected against physical attacks. The folder data, whatever their status are stored in the external NAND Flash (not hardware protected) and encrypted by using secret keys of type $k_U$. In the second case (a professional dealing with the folder of another user), the SPT must be able to serve as a communication channel between the central server and the folder it deals with. To do that, the SPT has to transmit into NAND Flash the recently updated data (Delta data) encrypted with one (or more) secret keys $k_D$, which are part of the administrative data carried by the SPT.

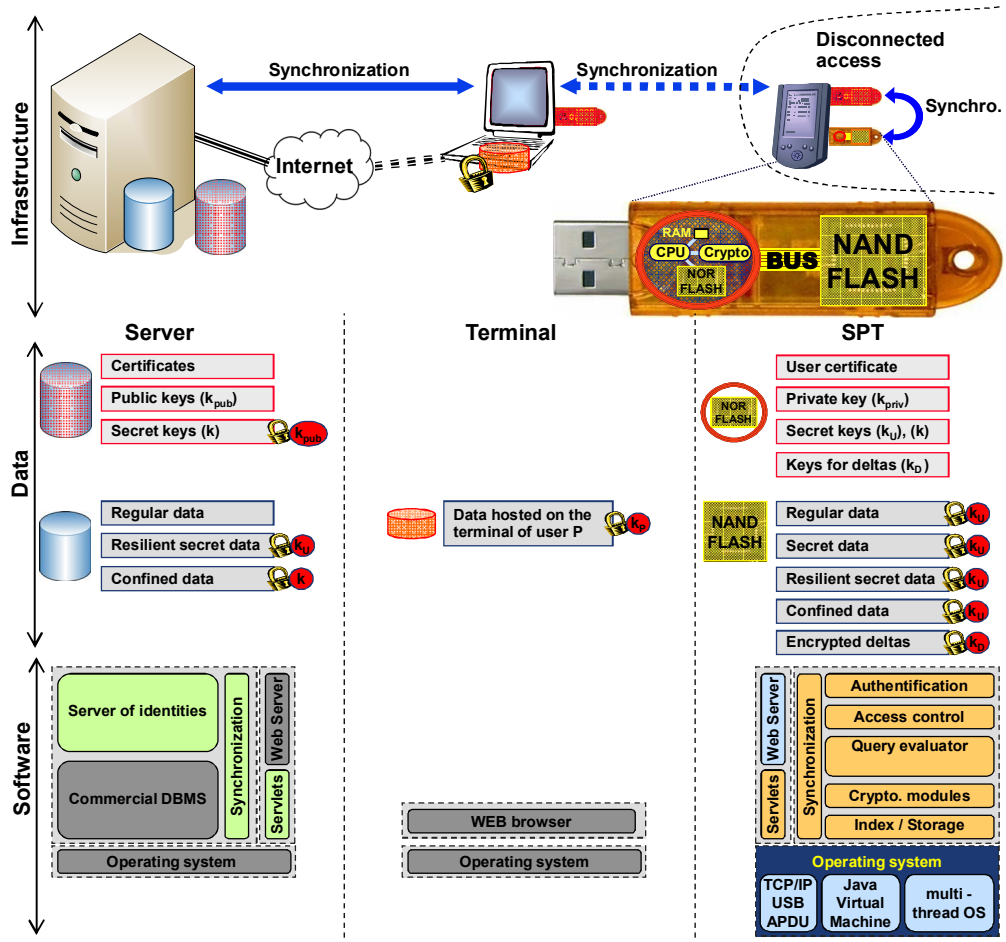


**Figure 1: Functional architecture**

# 4. Embedded database server

## 4.1. Functionalities

For the sake of conciseness, this section concentrates on the embedded database server, because this is the software component introducing the most challenging technical issues.

To manage secured and nomad folders, the SPT carries a full-fledged DBMS engine. This embedded DBMS: stores personal data on the external NAND Flash, builds and maintains indexes to make searches efficient, guarantees the atomicity of the update operations, evaluates SQL-like queries, has sophisticated access control enforcement (based on assertions, for example, only data satisfying assertion Q1 can be accessed by users satisfying assertion Q2). The access control mechanism is embedded in the secure microcontroller to be protected against any form of tampering (e.g., even if the folder is accessed through a spyware or virus prone terminal). The access control mechanism uses the query manager to evaluate the access control rules' predicates, and the query manager uses in turn the storage and index manager. Hence, the complete DBMS is embedded in the secure chip [10]. Unlike traditional smart cards storage memory, NAND Flash memory used by the SPT is not hardware protected against attacks and therefore must be protected by using cryptographic methods. These methods (encryption, hash, and version control) are used with a granularity and a cost which is compatible with the great number of random data accesses produced by the evaluation of database queries.

Finally, the synchronization module is also embedded in the SPT to synchronize the embedded folder with the copy stored in the central server, either directly when the SPT is connected, or through another SPT which is used as a communication channel (see scenario described in Section 2.2).

## 4.2. Technical challenges and solutions

The SPT framework introduces important new challenges [2]. Among which, the most difficult one is managing large databases on Flash with little RAM. To

tackle the RAM constraint, we designed and implemented a massive indexing scheme. This massive indexing scheme allows processing complex queries (with selection, projection, join and aggregate operators) over a large quantity of data (Gigabytes) while consuming as little RAM as possible and still exhibiting acceptable performances. The idea presented in [1] is to combine in the same indexing model generalized join indices and multi-table selection indices in such as way that any combination of selection and join predicates can be evaluated by set operations over lists of sorted tuple identifiers.

On the other hand, massive indexation causes a big problem in terms of Flash updates, due to the severe read/write constraints of NAND Flash (rewriting NAND Flash pages is a very costly operation). Therefore, we designed a structure which manages data and index keys sequentially so that the number of re-writes can be minimized. The use of summarization structures (based on bloom filters) and vertical partitioning reduce the cost of index lookups. These additional structures are also managed in sequence. A first implementation of this principle has been patented jointly by INRIA and Gemalto [11].

## 5. Conclusion

The functional architecture presented in Section 4 will be experimented in the context of a medical-social network providing medical care and social services at home for elderly people. The objectives pursued are (1) to re-establish a natural and powerful way of protecting and sharing highly sensitive information among trusted parties and (2) to build a shared medical-social folder providing the highest degree of availability, whatever the mode of operation (disconnected or not).

This project, partly funded by the Yvelines district and by ANR, involves the following partners: INRIA (the French National Research Institute in Computer Sciences), University of Versailles, SANTEOS (a French EHR provider), Gemalto (the smart card world leader), ALDS (a home care association) and COGITEY (a clinic for elderly people).

The design phase started in January 2007. A website dedicated to the SMSF project (DMSP) [12] provides detailed information about the project and the implemented prototype. The experimentation in the field will be conducted fall 2009. It should be conducted with a population of about 100 volunteer patients and 25 practitioners and social workers in the Yvelines district.

**Acknowledgments**

The authors wish to thank Laurent Braconnier and Jean-François Navarre (Yvelines District Council), Philippe Kesmarszky (ALDS), Sophie Lartigue (COGITEY), Morgane Berthelot (SANTEOS), Jean-Jacques Vandewalle (Gemalto), Georges Gardarin and Karine Zeitouni (University of Versailles) for their active participation in this project.

## 6. References

[1] N. Anciaux, M. Benzine, L. Bouganim, P. Pucheral, D. Shasha, "GhostDB: Querying Visible and Hidden Data without Leaks", *ACM SIGMOD Conference*, 2007.

[2] N. Anciaux, L. Bouganim, P. Pucheral, "Future Trends in Secure Chip Data Management", *IEEE Data Engineering Bulletin,* Vol. 30(3), 2007.

[3] R. Agrawal, R. J. Bayardo Jr., C. Faloutsos, J. Kiernan, R. Rantzau, R. Srikant, "Auditing Compliance with a Hippocratic Database", *Conference on Very Large Data Bases (VLDB)*, 2004.

[4] S. H. Brown, M. J. Lincoln, P. J. Groen, R. M. Kolodner, VistA, "U.S. Department of Veterans Affairs national scale HIS", *International Journal of Medical Informatics* 69, pp. 135-156, 2003.

[5] B. Blobel, P. Pharow, "A Model Driven Approach for the German Health Telematics Architectural Framework and Security Infrastructure", *International Journal of Medical Informatics*, Volume 76(2-3), pp. 169-175, 2007.

[6] European Directive 95/46/EC, "Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data", *Official Journal of the European Communities* of 11/23/1995 No L. 281 p. 31

[7] IPSOS Health / CNAM, "Opinions and Attitudes of Doctors and Patients about the Shared Medical Folder", 2003.

[8] InspireD Integrated Project, www.inspiredproject.com.

[9] G. Miklau, B. N. Levine, P. Stahlberg, "Securing History: Privacy and Accountability in Database Systems", *Conference on Innovative Data Systems Research (CIDR)*, 2007.

[10] P. Pucheral, L. Bouganim, P. Valduriez, C. Bobineau, "PicoDBMS: Scaling down Database Techniques for the Smartcard", *Very Large Data Bases Journal (VLDBJ)*, Vol.10, n°2-3. October 2001. Extended version of the paper rewarded by the Best Paper Award of the Int. Conf. on Very Large Data Bases (VLDB'00).

[11] P. Pucheral, S. Yin, "System and Method of Managing Indexation of Flash Memory", deposited by INRIA and Gemalto as a European patent n° 07290567.2-, May 2007.

[12] SMSF (DMSP) Project, www-smis.inria.fr/~DMSP.

[13] A. Westin, "Public Attitudes Toward Privacy and EHR Programs", *AHRQ Conference*, Washington, 2005.