SecMD

(Secure Medical Database)*

Pablo Ximenes[1], André dos Santos[1,2], Joaquim Celestino Jr.[2]

[1]Information Security Research Team (INSERT) – University of PR at Mayagüez (UPRM), USA

[2]Laboratório de Redes de Comunicação e Segurança (LARCES) – Universidade Estadual do Ceará (UECE), Brasil

pablo.ximenes@upr.edu, andre@dossantos.org, celestino@larces.uece.br

Abstract. *This paper describes an architecture to enforce security and privacy of patients' medical data called SecMD. A novel data representation scheme called Data Capsule (DC) is used to support such model. Data using DC's are represented as objects containing all the security parameters necessary to enforce a secure policy. The management of medical records, including their transfer from one entity to another (e.g., from Hospital A to Hospital B) becomes only a matter of managing objects. Security policies, auditing data, and all security enforcement data, which are part of a DC, are bound to the raw data, ensuring that security policies are always enforced.*

1. Introduction

Medical records have been historically a sensitive matter for the population. Citizens expect that their health conditions would not be publicly disseminated. A distrustful medical records system may deem patients insecure on revealing potentially harmful information. For example, in an attempt to avoid problems related to medical insurance, a patient could omit relevant information to care givers, compromising his or her own treatment. On another hand doctors and/or nurses need to have easy and prompt access to one's record in some situations (e.g., emergency and regular checkups). The question of who and how one can have access to somebody's medical records has usually been dealt with using pen and paper solutions: the person owner of the record (may be a representative with a power of attorney) signs consent documents to disclose the whole or part of the records to some other person. Ideally this solution could be effective. However, there are several situations that make this solution far from perfect. One example would be if the person storing the records has a financial incentive to leak that data. Another is the misplacement and/or improper storage of the records. The migration of information to digital data has made these potential problems even worse since digital data is easier to maliciously access and leak, and can be unintentionally misplaced, when compared with old paper storage solutions.

Electronic medical records (EMR) have recently received a great deal of attention not only from the population but also from legislative bodies of different countries. The European Union was one of the first to protect EMR by the Data Protection Act, published in 1988 and amended in 2003. The Data Protection Act is a general regulation that intends to regulate the access to any personal data, and not only EMR. The United States Congress, taking a different approach, signed in 1996 into law an act called

** This research was sponsored in part by the United States Army Research Office (ARO) grant number W911NF-07-1-0271.*

Health Insurance Portability and Accountability Act (HIPAA), which deals only with EMR. The HIPPA law provides the basic security guidelines that can be complemented by individual states with more restrictive laws. HIPPA was designed with EMR's in mind and may be a model of legislation to a number of other countries. In addition, the US represents one of the largest consumer market in the world and as such dictates many technologies commercially available. Therefore, it is interesting to further detail HIPPA and how technologies are addressing this act. From the perspective of data security, two major HIPAA's requirements deserve special attention: the Privacy and Security Rules. The Privacy Rule (effective 4/14/03 for most covered entities, and 4/14/04 for small health plans) intents to guarantee patient privacy by regulating how doctors, hospitals, healthcare plans, insurance companies, and other covered entities collect, manage, store, disclose, and utilize a patient's medical records. The Security Rule standards (effective 4/20/05 for most covered entities and 4/20/06 for small health plans) cover, among other things, technical security services (access control, audit control, authorization control, data authentication, and entity authentication) and technical security mechanisms (safe guards against unauthorized access to data by requiring integrity controls and message authentication, by requiring access controls and/or encryption, and, if network transactions take place, by requiring alarm reporting, audit trails, entity authentication, and event reporting) [1][2][3]. These strong requirements have had a direct impact on Data Base Management Systems (DBMS) for Medical records, as they must meet the basic criteria for compliance with HIPAA directives.

A myriad of DBMS products for managing Medical Records appeared in the market with promises of enforcing security and privacy requirements. Although some of these products use techniques such as multi-level security (MLS) [4], their implementations of the technique are over simplified providing a very coarse control (e.g., differentiating only administrative and non-administrative staff). Therefore, they cannot provide the privacy expected by a patient, who would ideally like to enable the access of information to others only on a need-to-know basis. For example, a patient may want to allow access to it medical records for an insurance company's physician only to items that may be requirements to his or her insurance application. On the other hand, he or she will be willing to give complete access to a personal physician, so he or she can conduct proper treatment. This way of authorizing access in different level, much in the same sense of the need-to-know model, is the key concept behind MLS.

Although products launched to address privacy concerns do not provide adequate security, large generic DBMS manufacturers have been incorporating general security features in their products for a long time. Potentially a Database Administrator (DBA) could implement methodologies and program the DBMS so that at least it would be close to address most privacy issues. One of the biggest problems with this solution is the reliance on this "excellent" DBA. Database administration in general is an area with a big lack of human resources. Adding to that, a DBA that could implement methodologies and program the DBMS in a way that preserves privacy of medical data requires an extensive training in Information Security, which by itself is also an area lacking qualified professionals [5]. Besides that, concentrating security enforcement on scarce technical personnel is a potential threat, since auditing inside attacks from such sources is a task that most usually can only be done by the very potential attacker. Quis Custodiet ipsos custodes? (Who shall guard the guards?) In fact, a common

misconception is that the main threat to privacy and confidentiality is from outside attackers. Several articles, including [6] and [7] cite internal attackers as a major risk. Often, a database administrator can be maliciously interested in obtaining medical records. Since he or she should have administrative access levels in order to proper administer the system, he or she will eventually be able to erase his own tracks, deeming almost impossible to audit a data leakage from this attack source.

Different institutions use different DBMS's to manage their data. Although one institution may be diligent in preserving patient's privacy, the transfer of Patients/Medical Records between institutions is far from trivial and may cause headaches for the diligent institution. Consider the scenario of transferring medical records between two hypothetical institution, Hospital A and Hospital B. Assume that Hospital A is very concerned with patients' security and privacy and has patient records managed by a powerful DBMS that uses several techniques for managing who has access to what data, including multi-level security (MLS). Let's assume the patient records have thorough information on patient's medical condition. A physician in order to assess patient's risk of going under surgery would have complete access to the patient's medical records. His or her role enables execution of a query that can show detailed information on patient's blood pressure, recent surgeries, and all sorts of information he or she might think are necessary to proper administer surgery. An insurance company representative would have limited access to some information on blood tests (previously authorized by the patient) in order to perform his or her evaluation on the patient insurance risks. The hospitals clerk can deal only with admitting information, but his role does not enable him to access diagnostic test results or anything else. All this is a reflex of Hospital A's concerns to patient's security and privacy and its respect to the law. Now let's assume Hospital A has to transfer the patient to Hospital B. How can one preserve authorities or privileges to access data if Hospital B manages database security by simple user access instead of MLS, or stores records using a primitive file system? Hospital A precaution could be deemed useless if, because the differences both hospitals have in dealing with medical records' security. This way liability for leakage of information is a major concern for Hospital A if it cannot trust the authorities or privileges to access data would be preserved after transferring the patient records. Thereafter, Hospital B would protect itself with lots of paper work that should be signed by the patient (or his or her power of attorney holder) and for Hospital B's representative. This would delay the process of transferring records and would still not prevent the compromise of medical records that could result in investigations and expensive law suits.

Another difficulty for providing security to Electronic Medical Records is the fact that many medical record systems that already are in production are implemented in such a way and use technology that makes them technically impossible to meet security and privacy requirements. Transitioning from legacy systems to meet new requirements isn't always cost effective and tends to present several technological barriers that many times results in a poor system.

This paper describes the Secure Medical Database (SecMD) Middleware; a software framework intended to enforce medical records security by means of the implementation of a novel approach for modeling data, the Data Capsule (DC) model. The DC approach enables data to be treated in an object oriented manner, allowing data objects to be treated in a safe, scalable, and user friendly way. Instead of raw data, a

data entity is composed of objects, with each object embodying features that can be used for accountability, decisions, and access control.

The SecMD Middleware sits between front-end applications and backend database management systems. SecMD allows interoperability of systems by abstracting medical records in data capsules that can be managed by any front-end application (as long as proper authentication is provided) and stored in any DBMS.

Database enabled applications usually rely on a separate software system, the DBMS, to manage their data. The communication between the software and its database inside the DBMS is done mostly through an industry standard language that reads, writes and manage a database within a DBMS through a set of structured queries that usually are transmitted through a network connection. This language is known as Structured Query Language (SQL) [8], and it is the standard for most DBMS vendors. Instead of a modification of a regular DBMS so the ideas described in this paper can be implemented, SecMD consists of a software abstraction layer (SAL) in the form of a network daemon (a software that provides network services) middleware that poses as Database Management System (DBMS), but it in fact is a proxy that intercepts database queries in SQL and manipulate them to enforce the DC based security model upon the actual DBMS. This way, any Medical Record Software can submit a query to SecMD as it was doing it to the actual DBMS. SecMD manipulates the query and transmit the resulting query to the actual DBMS, retrieves the results, and return them to the Medical Records application. This all is done with aims to enforce the DC based security model.

SecMD implements strong cryptography on the database in a way that a DBMS administrator does not need to decrypt any record, or if he/she does so it is done in a way that it does not compromise the security of medical records. The system also implements MLS in a manner that even if data is transferred between locations it preserves the same security properties for access privileges and authorization. This happens by embedding the access privileges and authorization structure onto the medical record itself, turning medical records data entries and its directives for access privileges and authorization into a single data object, the DC, that is stored encrypted in the actual database.

2. Related Work

Security has long been considered an important aspect for the implementation of effective electronic medical record (EMR) systems [9]. Several important studies that address this topic have been carried throughout the past decade. A particularly important study is the seminal work of Dr. Ross Anderson in 1996 [10]. Dr. Anderson proposed a security policy model specific to EMR's comparable to policies such as Bell-LaPadula [11] and Clark-Wilson [12], long known to designers of military and banking systems respectively. Dr. Anderson defined the theoretical foundations used for many of the current security approaches for EMR systems.

Even though the studies conducted on EMR security during the past decade has considerably advanced the state of art of the field, much of the work have investigated principles and theoretical considerations targeted to ad-hoc implementations. Such implementations cannot exchange data among them without reasonably effort on reformatting data and models. Very little research has been done for the interoperability of security mechanisms throughout a national EMR infrastructure. In fact, according to

[13] most of the current EMR systems fragment medical records by using means of storing and communicating data that are incompatible among the systems. Surprisingly, these incompatibilities are deliberate on several of the systems [13], aiming to prevent consumers from using alternate systems. This practice is a serious barrier for data sharing across different applications and institutions and thus has a direct effect on security interoperability. Even Europe that has been the pioneer on EMR mechanisms has scarce investigation on interoperability. Despite some initiatives on the use of smart cards and cryptographic mechanisms to build national EMR infrastructures [14], the European community now faces the same security interoperability challenges found in USA [15].

Commercial enterprises have leaded the efforts in implementing EMR systems motivated by the huge market segment created by HIPAA. However, most products are the results of adapting technologies that were already been used for solutions to other areas (e.g., banking and airlines) and do not address the peculiarities of the medical community. Recently, the number of academic studies on EMR systems security has increased, motivated by the several US initiatives supporting national use of electronic medical records. However, the recent research has concentrated on access models and anonymization techniques, leaving the important factor of interoperability unaddressed [16][9]. The lack of interoperability has deemed current security technologies for EMR systems inadequate focused [9] and insufficient for current needs [16].

Several studies support the need for a solid research effort on approaches that assure EMR security while allowing interoperability [9] [13]. The work presented throughout this paper addresses the interoperability challenges in a novel fashion that allows it to overcome the security issues related to allowing such interoperability. In fact, from the 6 major issues cited by Dr. Kenneth Mandl that must be addressed in order to proper develop an EMR infrastructure in the USA; four of them are in the core of the work described in this paper, namely: confidentiality, interoperability, accountability, and flexibility [13]. The other two, comprehensiveness and accessibility, although not directly addressed by the solution presented in this paper can use SecMD as an enabling infrastructure. In summary, despite the work herein described addresses problems that long have been under investigation (though it does that in a novel fashion), its main concern lies on the open issue of security interoperability, an issue that raises important research questions.

3. Data Capsules

SecMD uses a novel approach for modeling data called Data Capsules (DC). Although DC is a general approach and can be used in many more applications it is key component enabling the security and privacy of SecMD.

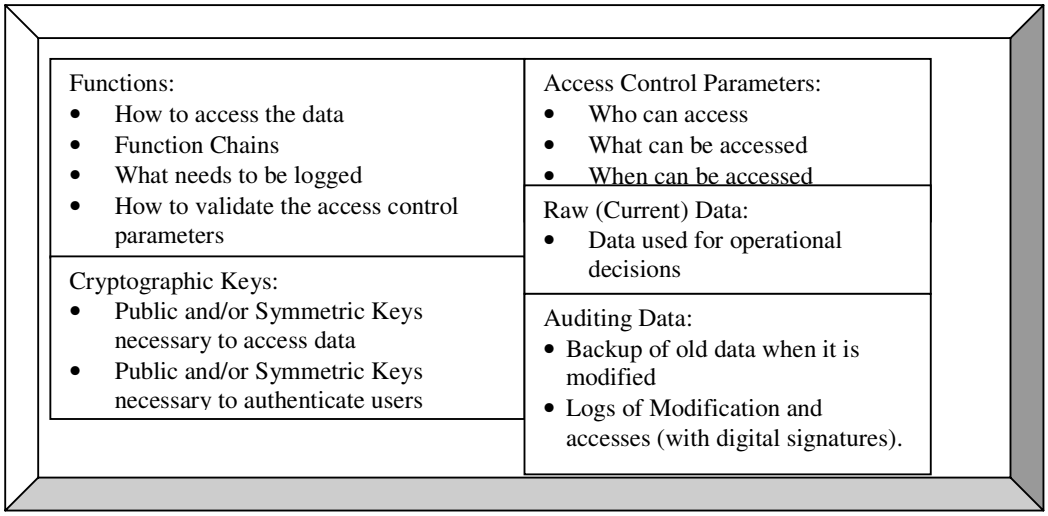

Figure 1. Data Capsule Example

A DC is a form of representing data in an object oriented manner in a way that security enforcement elements such as access privileges, authorizations, auditing trails, cryptographic keys, etc, are part of the data itself. Accessing the raw data contained in a DC consists in accessing the whole object, triggering all of its inner workings which are composed of protections techniques usually found outside the data structure. The protection of information and computation inside a Data Capsule are guaranteed by using mechanisms and/or specialized hardware (e.g., tamper resistant devices) to implement Trusted Computing Bases (TCB) [17]. Such implementation of TCB's, conceptual areas responsible for enforcing security policies, increases the overall assurance of the proposed system. An example DC is shown in Figure 1. A DC will generally consist of four basic elements that must be embedded into the capsule; these are functions, access control restrictions, auditing data, and raw data.

Functions. A computer system may interact with raw data in a data capsule only through functions specified and/or referred inside the DC. SysDefense proposes to use tamper resistant devices in some of its line of products to increase their assurance level. Functions running inside the tamper resistant devices will enable the enforcement of complex security policies whenever the data is accessed in a very flexible manner. The flexibility provided will allow policy changes by changing functions inside DC's, enabling updates as the need arises.

Access Control. Access control is performed by the functions that are encapsulated with the data according to what is established by access control parameter inside the DC. For example, a set of basic functions implementing authentication and access control list authorization will be present in a DC and will use access control parameters to authorize an action. An authentication function may use cryptographic keys (an access control parameter) to authenticate a user.

Auditing Data. The DC approach binds data with its auditing trail. Simply by reading a DC, one is generating auditing trail that later may be used to track potential harmful activities.

4. Architecture

SecMD's architecture consists of a middleware that sits between existing Database Management Systems (DBMS) and data management front ends. The middleware intercepts database queries and manipulate them to enforce the DC based security model upon an actual DBMS. This architecture is modular and consists of the modules shown in Figure 2 and described later.

4.1. Database Module

This module is responsible for the interface between queries from the Client Software to the actual DBMS as well for all techniques used to maintain DC's in the actual DBMS. The Database Module interfaces directly with the client, receiving all database queries and responding to them. Queries from the Client software are mapped to queries for DC's that are passed to DBMS. The DC's returned by the DBMS are then forwarded to the DC module for interpretation and security policy enforcement. Both the modified DC (at least one audit entry is inserted at each access) and the result of the Client Software query are returned to the Database Module; which in turn sends the first to the DBMS and the later to the Client Software.

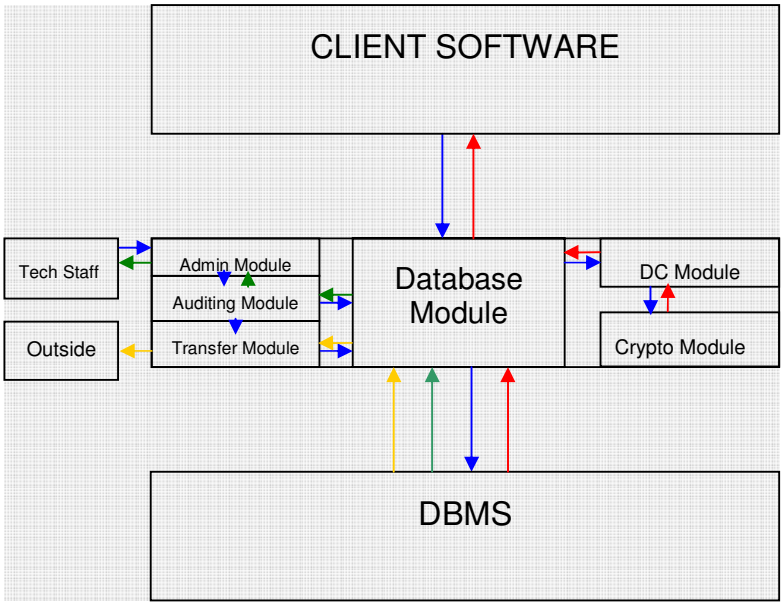

Figure 2. SecMD Architecture

4.2. DC Module

This Module is responsible for implementing the trusted computing base necessary to protect DC manipulation. A DC is interpreted inside this module; the security enforcement mechanisms are applied; and data may be returned. The security enforcement mechanisms include verifying all access privileges and authorizations, updating auditing trails, and executing inner functions.

The DC module can be implemented either as a shared resource, where many DC's are interpreted, or as a private resource, where only one DC is interpreted. The DC module may store parts of or the whole Data Capsule, limited by its storage space and general

policies allowing such use. As an example, an individual DC module can be used to carry a DC that includes the whole medical history of a specific patient. Although all or part of the data are also present in the DBMS of the entity (hospital, doctor) that has interfaced with the patient, this DC is not dependent of any DBMS and is a self contained, portable, health history of the patient.

4.3. Cryptography Module

SecMD architecture uses cryptography extensively to guarantee security properties. The Cryptography module is responsible for all encryption within the system including all cryptographic keys management. It is responsible for the encryption and decryption process, key generation and key management. The Cryptography Module can interface with external hardware device such as crypto boards in order to optimize cryptographic procedures.

4.4. Administrative Module

This module is responsible for system administration, interfacing with the administrative technical staff. A Graphical User Interface (GUI) and a command shell are implemented to allow administrators of the system to interact with this module. An administrative user can use an interface to manage the system in its whole, including managing user and roles, checking auditing trails, and managing policies. An administrative user is also able to override security policies like resetting passwords and querying unauthorized DC's.

An administrative user has a very powerful control of the system. Because of that both authentication of administrative users and auditing trails of their actions must be carefully implemented. Nothing can stop an administrative user from tampering with the whole system if no logging of her actions is generated (or can be deleted). SecMD implements strong authentication procedures and auditing logs to prevent and identify misuse of the system. SecMD uses tamper-resistant devices like smart cards to enforce authentication and guarantee tamper-resistant auditing trails.

4.5. Record Transfer Module

This module is responsible for preparing Medical Records for transfer between institutions. Basically, it gathers DC's in such a way that transfers are safe-guarded and DC's inner security requirements are met on the destination institution.

5. Performance Evaluation

The main performance bottlenecks for the proposed system are the cryptography overhead and the log data growth rate.

Based on the findings in [18] and on the statistics of the "US 2005 National Hospital Discharge Survey" [19] we were able to model the general database access patterns of the proposed system.

According to [18] the overhead added by the cryptographic operations that are to be part of a Medical Records DBMS in order to make it HIPPAA compliant accounts for less than 7% of service degradation in the worst case, with average of 5%. We assume that is an accepted level of service degradation, especially because the times acquired by [18] in order to reach this figure for service degradation were done with no aid of specialized

cryptographic hardware. Such hardware if added to the proposed scheme would decrease these overheads even more.

This way, our main concern is the growth rate of the log padding of data capsules. In order to analyze that, we have used the hospital workflow model devised in [18] together with some of our own observations. This way we modeled the basic impact of a hospital visit over the proposed system. Table 1 shows the impact in log padding growth for each stage of a hospital visit. We can see that in average each procedure adds approximately 23 bytes of data to a DC's log padding.

Table 1. Impact of Visit over Log padding per workflow stage

Workflow Stage	Log padding (in bytes)
Registration	20
Notify Hospital Information System (HIS) of Visit	20
Triage	20
Schedule exam	20
Conduct Patient exam	30
Patient Report Generation	30
Report transmitted to HIS	20
Total	160

In order to find the overall daily impact of the log padding in a major hospital we have derived the sample data in [19] which accounts 375000 discharges distributed over 444 hospitals. That averages 845 visits each day. Even though this number expresses the particular reality of the USA, we are confident that these figures give the necessary empirical evaluation about the overall performance of the proposed system. This way we have contrasted the findings of table 1 with the simulative number of 845 hospital daily visits which can be seen in table 2.

Table 2. Impact of all visits in a day over Log padding per workflow stage

Workflow Stage	Daily Log padding (in bytes)
Registration	16900
Notify Hospital Information System (HIS) of Visit	16900
Triage	16900
Schedule exam	16900
Conduct Patient exam	25350
Patient Report Generation	25350
Report transmitted to HIS	16900
Total	135200

As we can see in table 2, the total daily database growth overhead due to log padding for the proposed scheme averages 132 KB. This means that in the course of a year, simple visits in a major hospital will demand an update of only 50 MB of storage space in average. Due to the increasingly low costs of data storage devices, we consider this overhead acceptable.

6. Conclusion

We have presented the initial results of a feasibility study for the Secure Medical Database (SecMD), a database middleware architecture that applies the novel concept of Data Capsules (DC). We have shown the general architecture of the proposed system and demonstrated that it presents acceptable levels of performance if it is to be deployed as a full fledged application in a major hospital. As future work we intend to refine the

representation of medical records as DC's, to design an abstraction layer between DC's and general DBMS's, and to deploy prototype security analysis and benchmarking.

7. References

[1] US Department of Health and Human Services – DHHS (2006), Public Law 104-191, "Health Insurance Portability and Accountability Act of 1996," Available in Online: http://aspe.hhs.gov/admnsimp/pl104191.htm

[2] A. M. Snyder (2003), "Performance Measurement and Workflow Impact of Securing Medical Data Using HIPAA Compliant Encryption in a .NET Environment", Master Thesis, University of Virginia, USA.

[3] HIPAA Advisory (2006) "Status of HIPAA Regulations Compliance Calendar", Available Online in: http://www.hipaadvisory.com/regs/compliancecal.htm

[4] R. J. Feiertag, K. N. Levitt, and L. Robinson (1977), "Proving multilevel security of a system design", ACM Symposium on Operating Systems Principles, P. 57 - 65, ACM Press New York, NY, USA

[5] President's Information Technology Advisory Committee (2005), "Cyber Security: A Crisis of Prioritization", Available online at: http://www.hpcc.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf

[6] R. Simpson (1996) Security Threats are Usually an Inside Job, Nursing Management, 27(12): 43.

[7] T. Rindfleisch (1997) Privacy, Information Technology and Health Care, Communications ACM, 40(8): 93-100.

[8] T. Connolly, C. Begg, and A. Strachan (1998), "Database Systems: A Practical Approach to Design, Implementation, and Management", Addison Wesley Publishing, USA

[9] Don E. Detmer (2003). "Building the national health information infrastructure or personal health, health care services, public health and research". BioMed Central Medical Informatics and Decision Making, 3(1):1.

[10] Ross J. Anderson (1996). "A Security Policy Model for Clinical Information Systems," in Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pp. 30–43, IEEE Computer Society Press, Los Alamitos, CA

[11] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations and model. Technical Report MTR 2547 v2, MITRE Corporation, 1973.

[12] D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security policies. In IEEE Symposium on Security and Privacy, pages 184{194, Oakland, April 1987.

[13] Kenneth D. Mandl, Peter Szolovits, and Isaac S. Kohane (2001) "Public standards and patients' control: How to keep electronic medical records accessible but private," British Med. J., vol. 322, pp. 283–287.

[14] Roderick Neame (1997). Smart cards: the key to trustworthy health information systems. BMJ 1997;314:573–7.

[15] Gérard Comyn (2006) "Connected Health: Quality and Safety for European Citizens", Report of the Unit ICT for Health in collaboration with the i2010 sub-group on eHealth (formerly known as the eHealth working group) and the eHealth stakeholders' group.

[16] Khin T. Win (2005). "A review of security of electronic health records. Health Information Management"; 34(1): 13-8.

[17] André dos Santos (2000), "Safe Areas of Computation (SAC) for secure computing," PhD dissertation, University of California Santa Barbara, Online at http://ece.uprm.edu/~andre/Dissertation.pdf.

[18] Snyder, Andrew Morgan (2003) "Performance Measurement and Workflow Impact of Securing Medical Data Using HIPAACompliant Encryption in a .NET Environment," Master's thesis, University of Virginia, USA

[19] DeFrances, Carol J. and Hall, Margaret J. (2007) "2005 National Hospital Discharge Survey", Advanced Data from Vital and Health Statistics #385, US Center for Disease Control and prevention (CDC), Division of Health Care Statistics, USA