Journal of Mobile, Embedded and Distributed Systems, vol. III, no. 2, 2011
ISSN 2067 – 4074

Smart Cards Applications in the Healthcare System

Claudiu Oltean
Computer Science Department,
IT&C Security Master
Cybernetics and Economic Informatics, Bucharest Academy of Economic Studies
Calea Dorobantilor Nr. 15-17, Room 2315, Sector 1, Bucharest
ROMANIA
claudius83@gmail.com, http://ism.ase.ro

Abstract: Current medical system based on medical records and health books is outdated and no longer meets the new requirements. Essential information security in terms of data privacy, integrity and authenticity, is not assured. Healthcare fraud with medical records is quite easy, because there is no security features to prevent this. Obtaining prescription drugs is slowly, the patient is forced in most cases, to go to the pharmacy staff to get their prescription. Another issue is data portability because each clinic can use a proprietary format of medical records, which is not always standardized. Modern and efficient healthcare system can be achieved by introducing smart cards and related software. Their introduction in addition to the portability and data security, reduce costs for both patient and medical institutions. The result will be increase confidence and patient satisfaction in medical institutions. Developed software package includes software applications which manage medical archive to smartcard, in a secure form and a software module which can be used for e-commerce transactions. All developed software application meets current standards for data security. Implementation of such solutions in practice would significantly reduce current costs in healthcare system.

Key-words: Gemalto .NET Smart Card, CCR-Continuity of Care Records, SET-Secure Electronic Transaction, ASN1, XML, SMHApp, SETApps, AES, Rijnadel, Remoting

1. Introduction

Current healthcare needs to be improved because it is based on old data storage techniques. Nowadays technology allows the upgrade and the benefits would translate into lower costs and improve quality of medical care. Upgrading and improving existing medical system can be made by introducing smart cards and related software applications that meet the following requirements:

- improving health services and reduce costs for both patients and medical staff;
- ensure portability of medical records for patients;
- ensuring data security, authentication, confidentiality, integrity and authenticity;
- introducing a system for purchasing prescriptions through specific e-commerce transactions;
- prompt access to emergency medical information;
- reducing medical fraud.

This is a post conference paper. Parts of this paper have been published in the Proceedings of the 3rd International Conference on Security for Information Technology and Communications, SECITC 2010 Conference (printed version).

2. Current problems and solutions

2.1. Data security

One of the current requirements in healthcare system is the security, confidentiality and authenticity of medical records. Since patient related data are not encrypted, confidentiality is not ensured because each person who has access to medical records can see patient private data. Forgery is also fairly easy to achieve, because the doctor's signature and seal are not a very good security feature. Storing medical records on the smart card by replacing traditional medical records based on papers resolves the security issue. Confidentiality is ensured by encrypting the data, authenticity plus

integrity by using electronic signatures and data access management by introducing access rights policy. More over smartcards are cheap and offers quick and easy access to medical records.

2.2. Portable medical records

Currently physician writes manually medical data in to medical records. Very often this information is not very clear and legible which generates many problems with reading and interpretation. Some extreme cases may even lead to confusion, which would have very serious consequences for the patient. Smart card can store the entire medical archive and many other important information for a person. Portable medical records can be achieved only by adoption of standards. At present there are two standards for the portable medical records: *HL7*, *CCR* and *CCD*.

Health Level 7 (HL7) is a standard approved by the *American National Standard Institute (ANSI)* used mainly to change the medical and administrative information between healthcare organizations using different software.

Standard Continuity of Care Records (CCR) was defined jointly by *ASTM International*, *Massachusetts Medical Society*, *Healthcare Information and Management Systems Society, American Academy of Family Physicians and American Academy of Pediatrics.*

Its goal is to organize and improve transport of medical information, reduce medical errors originating from incorrect information or lack of medical care and to improve patient healthcare by reducing its role in providing information to medical personnel.

Continuity of Care Document (CCD) is a newer standard that would replace the CCR and HL7 standard.

It is a result of collaboration between HL7 and CCR standards. Basically CCD standard was created for institutions already using HL7 and is adapted to meet the requirements of CCR standard.

2.3 Getting prescription using E-commerce electronic transactions

Currently, medicines can only be taken from the pharmacy only based on a prescription. A payment system with smartcard would solve this problem, patient being able to order the prescriptions. Basically patient with a personal computer and a smartcard inserted in to a card reader is authenticated using a PIN number and based on his medical records, can order medicine from the pharmacy. The order is then processed by the pharmacy server and then medicines will be delivered to the patient. This procedure is similar with all e-commerce transactions.

The advantage is that the patient is not forced anymore to go to the pharmacy. In addition this is a great advantage for those medicines which are hard to be found and only in a few pharmacies. For the patient it doesn't matter what pharmacy has the medicine since the whole process is automated.

2.4 Reducing administrative costs

Reducing costs is one of the major issues in the health care system. Paper consumption is used to fill very large medical data. More than that, a doctor uses almost half the time just to fill data in healthcare records. By using smart card, paper consumption would be reduced greatly, and reduce costs should be obvious. Also use a software application will significantly decrease the time needed to fill medical data.

2.5. Reduce fraud

Fraud in health care affects everyone, even if a small percentage of people frauds the system. The bigger the fraud is , medical service is worse. Examples of fraud are: charging for medical services not actually provided, falsifying a patient's diagnosis to justify various medical services which are not needed, the charging of a more expensive service than actually provided, use of other person insurance card, forging prescriptions. To prevent fraud, in the doctor's office it can be used a computer connected to a card

ISSN 2067 – 4074

reader. The computer runs a software application with which the doctor saves patient information in to the smartcard. Reading or writing the smartcard is made securely, because each time for accessing the card introducing a PIN number is mandatory. Based on stored medical records, pharmacy can issue medicines to patient. In addition, at the doctor's office or pharmacy, it can be sent a secure message to the central database, to record the medical service provided. Fraud is eliminated because the data stored on the smartcard cannot be modified, and for each medical service provided, a secure message is sent to central database to record the service. In addition medical staff can verify at any time patient insurance.

2.6. Quick access to emergency medical information

In emergency situations such as accidents where the person is unconscious, first aid physicians would need critical medical information like blood type or allergies. This critical information can be stored in a special area of the smartcard, without the need for patient consent, to be read. Moreover doctors may send this information directly to the hospital so the hospital's medical staff will be ready when the patient arrives. Storing information on smart card can be very useful for elderly people or foreigners who do not speak the language of that country. In this case the doctor no longer has to interrogate people to get medical information.

3 Hardware and software solutions for the healthcare system

3.1. Gemalto .NET Smart card

A smart card is just like small mini computer with a design similar with a normal magnetic card. It contains an area of memory used for storing data and a microprocessor used most often for cryptographic operations. The difference with an ordinary magnetic card is that a smart card contains a microprocessor.

3.1.1 Advantages and benefits

Gemalto .NET smart card is built following the *ISO 7816-1* standard, by the company *Gemalto* leadership in digital security. The figure below shows an Gemalto .NET smart card.

From a programmer point of view, software development is easier and faster than other smart card because the card is based on *Microsoft .NET Framework*.

Figure 1. Gemalto .NET smart card [16]

In addition, a programmer can choose one of the programming languages *C#* or *Visual Basic* to develop its solutions. Card management can be made with the utility *Card Explorer,* which can manage easily the entire card.

3.1.2 Concepts and Designs

Executable files are similar to conventional libraries generated using the Microsoft. NET Framework, with de difference that generated files for the smart card, go through a conversion process to optimize and to reduce size and ensure compatibility with the data types supported by the card [3]. This process is hidden from the user. Also executable files are signed using the public key stored in the manifest file associated.

Application domain concept allows to run simultaneous multiple independent applications on the smart card [3]. Practically every application runs in its own associated domain. In addition data integrity is ensured because access to data from another domain is not possible.

Remoting technolgoy is the main concept used in Gemalto .NET smart card. *Remoting* is aimed at communication between application domains and processes or different machines [3]. It allows access to different application domains. Communication between a client

and a server running on the smart card is done using *Remoting*. Communication is independent of transport protocol which can be *TCP, HTTP, or APDU*.
Another useful concept is the usage of *Sinks* [3]. Using Sinks is useful only if it is important how data are sent to smart card from the client software. One such case is when we want to encrypt data sent to smart card, to ensure privacy.
Gemalto .NET smart card has also a *Garbage Collector* [3]. Memory management is done automatically by the *Common Language Runtime* and memory is deleted automatically without the need of explicit intervention of a programmer. When a variable is referenced Garbage Collector deletes memory for that variable.
The card supports conducting operations in *one transaction* which ensures data integrity [3]. When a transaction is created it preserves the original data, and the change is done, only if the transaction is completed successfully. Any method can be marked with the attribute *[Transaction]*.

3.2 Continuity of Care Record (CCR) XML data structure

Generally, each patient's medical data are stored internally by each medical staff. Over time a patient can visit different doctors and the result is a dispersion of medical data in different places and an incomplete medical archive. In this context portability of medical data is essential. There are several standards for the portability of medical data, each of them based on XML files. Developed software package is using *Continuity of Care Record (CCR) XML* standard model to ensure portability [2]. XML file contains tags for each part of health information.
CCR XML file consists of three parts: *Header, Body, and Footer*.
The *Header* contains tags which identifies the document id, langguage, version, creation date, purpose and patient id. Table 1 contains an example of CCR Header tags.

Table 1.CCR Header

<CCRDocumentObjectID>
...
</CCRDocumentObjectID>
<Language>...</Language>
<DateTime>...</DateTime>
<Patient>...</Patient>
<From>...</From>
<Purpose>...</Purpose>

The *Body* contains tags for entire medical records like problems, prescriptions, results, procedure etc. Table 2 contains an example of CCR Body tags.

Table 2. CCR Body

<Body>
<Payers>...</Payers>
<Support>...</Support>
<FunctionalStatus>...</FunctionalStatus>
<Problems>...</Problems>
<FamilyHistory>...</FamilyHistory>
<SocialHistory>...</SocialHistory>
<Alerts>...</Alerts>
<Medications>...</Medications>
<Immunizations>...</Immunizations>
<VitalSigns>...</VitalSigns>
<Results>...</Results>
<Procedures>...</Procedures>
<Encounters>...</Encounters>
<PlanOfCare>...</PlanOfCare>
</Body>

The *Footer* includes tags for references, electronic signatures, and id's for all actors used in the document. Table 3 contains an example of CCR Footer tags.

Table 3. CCR Footer

<Actors>…</Actors>
<References>…</References>

ISSN 2067 – 4074

<Signatures>…<Signatures>

3.3 High level design and structure for the developed software solution.

Developed software package attempts to solve the main problems outlined above. To use this package it is needed at least one Gemalto .NET smart card, a card reader and a personal computer with Microsoft .NET Framework 2.0 installed.

The software package contains two applications *SMHApp* and *SETApps* developed and implemented in the *Microsoft Visual Studio C#* environment. *SMHApp* is responsible for the management of personal and medical data on the smartcard. Through it you can add, delete or modify medical records in to a Gemalto .NET smart card. *SETApps* is a suite that implements the *Secure Electronic Transaction (SET) e-commerce* protocol [8]. The purpose of this package is to enable patients to order medicines without being forced to go to the pharmacy. Software package can also be used to order any type of products, the only difference being that instead of health insurance and prescription recipe, personal banking data is used.

3.3.1. SMHApp (Smartcard Health Application)

It is a graphic application that displays and modifies personal and medical data on a Gemalto .NET Smart card. Data security is ensured by encrypting with a symmetric *AES Rijnadel* key and authenticity and integrity by using electronic signature. The key is obtained from a password that is required each time patient or medical staff wants to access the smartcard. Figure 2 shows application architecture and its two components SMHAppClient and SMHAppServer.

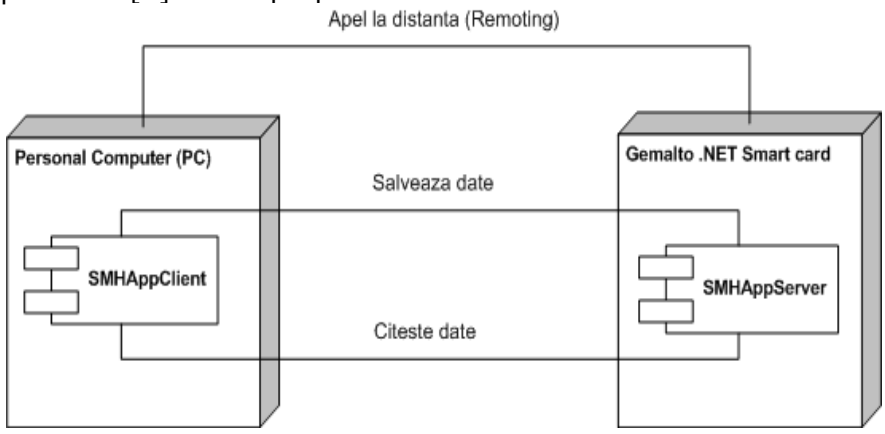

Figure 2. SMHApp Architecture

SMHAppClient is a friendly user application which cans displays and modifies medical record and other data which are stored on smart card. From development point of view its role is to package this information in an XML format, to archive using GZip algorithm, and to send it to SMHAppServer module. It can package three types of data: personal data (like name, age, date of birth, address), e-commerce data (banking numbers) and medical archive. Medical archive is stored in a *XML* file compliant with CCR xml document [1]. It also computes electronic signatures for the data stored in the medical archive. *SMHAppServer* is a software application that runs in to smart card similar with a Microsoft Windows service. Its role is to liaise with SMHAppClient card application. Only through it SMHAppClient can access data stored in to the card. All data manipulation is made through this service. For each section of the data presented above SMHAppserver make an *AES* symmetric encryption, and then saves the data to a file. In this way data confidentiality is ensure. Communication between SMHAppClient and SMHAppServer is made using *Remoting* technology [2]. This is actually based to *Remoting* technology used in *Microsoft .NET Framework* with the difference that *APDU*

protocol is used, and not *TCP/IP*. However as a programmer it is not require knowing APDU protocol, because *Remoting* is an abstract technology and does not depends on the protocol used. If the protocol is changed let's say to TCP/IP, it becomes a simple client-server application that can be debugged without needs of the smart card. Gemalto Smartcard. NET contains a compact version of Microsoft. NET Framework required for running applications on the card, and has all the important cryptographic operations. These two applications are developed using Gemalto .NET development kit, version 2.0.54.

3.3.2. SETApps (Secure Electronic Transaction Applications)

SETApps is a module which implements basic functionality of SET protocol [8]. Currently due to some limitation of the smartcard there is only a PC version but the major objective is to port this module to smartcard. In this way smart card can be integrated with e-commerce protocols which will increase security.

The purpose of this package is to provide a way to order medical prescription without being forced to go to the pharmacy. Software module can also be used to order any type of product, not only prescriptions, the difference being that instead of health insurance and medical records, banking numbers are used.

SET protocol implies the existence of the following entities:

- Cardholder - The person who holds a card that can be used in the electronic commerce.
- Merchant - An entity that provides goods and services in the e-commerce
- Issuer - Entity issuing the cards used in electronic commerce
- Acquire - Entity used by Merchant for payment processing.
- Payment Gateway - Acquire interface for payment processing.
- Certificate Authority - Entity that issues digital certificates.

SET protocol is developed according compliance requirements described in *Computer Security Group, University of Cambridge Computer Laboratory, MasterCard and Visa specifications based on 8 August 1996* [8]. However it is implemented only the part between Cardholder, Merchant and Payment Gateway. Communication is done through ASN1 DER encoded message. Each message has an equivalent C# class that packs the data and then converts it into a DER format. The package contains SETApps core library, which implements SET protocol, and client-server demo applications CarholderApp, MerchantApp, PaymentGatewayApp. Part of the protocol is implemented between the Cardholder, Merchant and Payment Gateway, but in a real scenario all the parts of the protocol must be implemented. Figure 3 shows the architecture of SETApps.

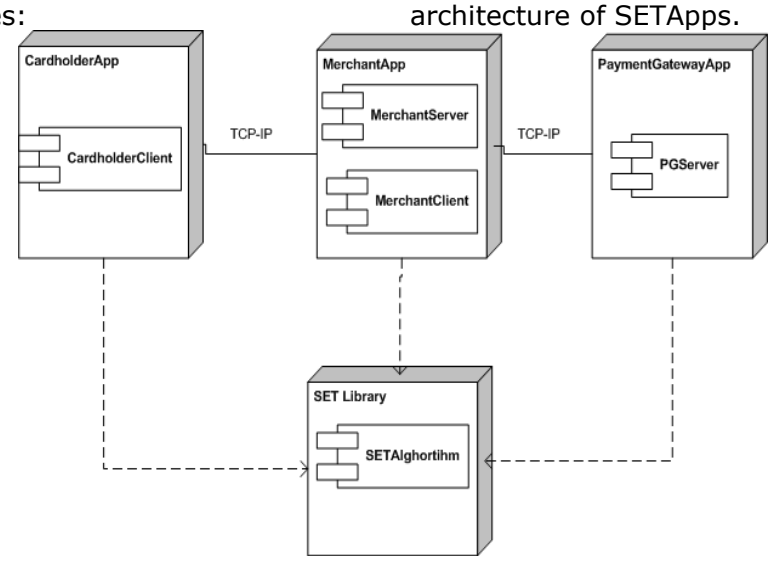

Figure 3. SETApps Architecture

ISSN 2067 – 4074

CardholderApp is a client application designed for the cardholder. Its main functions are related to connecting at MerchantApp server and display all messages sent or received in the SET protocol.
MechantApp is a client-server application. It contains a server-side module that connects to CardholderApp and a client-side module through which the application connects to PaymentGatewayApp.
PaymentGatewayApp is a server application design for Payment Gateway. It contains a server-side module through which MerchantApp connect to it. It displays also all the SET messages sent or received.
SETAlghorithm is the core library that implements the SET protocol. It is used by each of the applications listed above. The goal is to port this library to smart card, so that operations can be made secured inside the card. The messages are packed as ASN 1 DER, according to compliance specified in the SET protocol. However since the purpose is to demonstrate the advantage of using smartcard in health care system, this set of software applications implements only the most important aspects of the protocol, especially those who are related to security.
SET message described in ASN1 language files have been parsed with *ASN1C* utility to generate C# classes [14]. Generated messages are built in compliance with PKCS7 standard for cryptographic messages. SET Message are build only in to SETApp library, because the purpose is to make the library application independent. Since SETApp is a library that is distributed as a DLL file it can be easily integrated into any application that uses Microsoft software development technologies.

4. Conclusions

Through this study were presented some practical solutions that would solve some of the major issues in the healthcare system by use of Gemalto .NET smart card, CCR standard for data portability and encryption plus electronic signature for data security. More than that it was summary implemented SET protocol module needed for ordering medical prescription.
Developed software package is a general one and for an implementation in practice it needs to be customized for each client.
SMHApp application stores medical records in to the smart card in form of XML files. This is not effective from the storage perspective and maybe a smaller format needs to be use like ASN1. In this case to achieve compatibility with CCR standard a conversion method form ASN1 to CCR XML should be implemented.
Uses of ASN1 for storing the archive would also eliminate archiving the medical records before saving in to smart card.
SETApps library module needs to be port to Gemalto .NET smart card because currently only a PC version is available.
The advantage of using smart card compared to a personal computer is the fact that the smart card security level is much higher. Upgrading the current medical system can be done only through its computerization and the applications listed above may be a good starting point.

References

[1] Smart Card Alliance, Smart *Card Applications in the US Healthcare Industry*, www.smartcardalliance.org, February 2006, HC-06001

[2] ASTM INTERNATIONAL, *Continuity of Care Record*, www.astm.org.

[3] Gemalto, *NET Card Technology Overview*, NETSmartcardOverview.chm

[4] Gemalto, *.NET Smartcard Framework Documentation*, NETSmartcardFramework.chm

[5] Gemalto, *.NET Smartcard Framework Off Card API*, NETSmartcardClientAPI.chm

[6] Gemalto, Gemalto *.NET v2/v2+ Smart Card User Guide*,

www.netsolutions.gemalto.com, July 2008

[7] Gemalto, *Tehnical Specifications*, www.gemalto.com/products/dotnet_card

[8] Computer Security Group University of Cambridge (1996), *SET Protocol Description*, http://www.cl.cam.ac.uk/research/security/resources/SET/intro.html

[9] Cristian TOMA, *Security in Software Distributed Platforms*, ASE Publishing House, Bucuresti, 2008, ISBN 978-606-505-125-6

[10] Ion IVAN, Cristian Toma, *Informatics Security Handbook 2nd Edition*, ASE Publishing House, ISBN 978-606-505-246-8

[11] Victor Valeriu Patriciu, *Lecture Securitatea plătilor si comertului electronic* (in Romanian), 2009

[12] Jay Hilyard, Stephen Teilhet, *C# Cookbook, 2nd Edition*, O'Reilly Publishing House, January 2006, ISBN 978-0-59-610063-6

[13] *Open eHealth*, SampleCCRDocument.xml, http://gforge.openehealth.org/gf/project/ipf/scmsvn/trunk/ipf/modules/cda/src/test/resources/SampleCCRDocument.xml

[14] Objective SYSTEM Inc, *ASN1C*, www.obj-sys.com

[15] Wikipedia, Smart Card, www.wikipedia.org

[16] Wikipedia, SmartCardPinout.svg, www.wikipedia.org/wiki/File:SmartCardPinout.sv