An architecture for secure e-Health systems

Dario Salvi, Elena Villalba Mora and Maria Teresa Arredondo Waldmeyer
Universidad Politecnica de Madrid
Madrid, Spain
Email: dsalvi,evmora,mta@lst.tfo.upm.es

***Abstract*— The aging of population in developed countries and, consequently, the growth of chronic deseases is pushing public health industry to find solutions in order to bring healthcare to patients homes. This can significantly contribute to the reduction of healthcare cost by avoiding unnecessary hospitalisations and ensuring that those who need urgent care get it sooner.**

Many e-Health systems for remote patients assistance are being developed among the research community. Security has become a key issue in order to ensure the privacy and confidentiality of individuals.

This paper proposes a general architecture for e-Health systems and discusses the security issues it leads to and provides a specific framework based on interoperability and well-known standards.

I. Introduction

Continuous monitoring of physiological and physical parameters is necessary for the assessment and management of personal health status.

Delivery of healthcare services beyond the enterprise level to the regional, national or cross-border area places new challenges for security implementation. A common security framework is needed for national and pan-European e-Health and telemedicine applications.

Within the European legal framework there are two laws directly affecting the data management in e-Health environment.

The former, the Directive 97/66/EC ”concerning the processing of personal data and the protection of privacy in the telecommunications sector”, is the one that gives the main guide-lines of data protection. As a general rule, the directive establishes that sensitive data can only be processed with the explicit consent of the individual, except in sepcific cases such as where there is significant public interest [1].

The latter is the Recommendation No R(97)5 ”on the protection of medical data” which focuses explicitly on the subject of this paper [2]. The law defines the general rules for the acquisition, the storage and the treatment of the data of a patient, and the rights and duties of each of the involved actors.

As regarding the roles of the people comprised in the acquisition of personal data, the general rules of the recommendation can be summarized as follows:

- Medical data should be collected only by healthcare professionals or
- by individual bodies working on behalf of healthcare professionals
- Controllers of files who are not healthcare professionals are subject to rules of confidenciality comparable to those incumbent upon healthcare professionals

The need arises to define, in an unique way, what can be defined as medical data, and in particular what can be defined as *personal data*.

The Recommendation states that personal data is any information regarding an identifiable person, for example the name, the surname, the birth date etc. On the other hand, anonymous data is the one that does not allow identifying a person, for instance an ECG, or an average blood pressure, or a weight.

However some kind of data would result in a sort of a grey area, like the race, the country or partial genetic information. The joint set of data concerning a person will define its level of *anonimity* as far as it gives the possibility, with reasonable resources, to find out the person the data are related to.

In order to allow personal data to be protected, the recommendation gives some guidelines. Summarizing, and focusing on the ones resulting more interesting in our opinion:

1) Appropriate measures for confidentiality, integrity and accuracy of data must be taken.
2) Appropriate measures of access control must be granted.
3) It must be enabled the separation of:
   - Data relating to the identity of persons
   - Administrative data
   - Medical data
   - Social data
   - Genetic data
4) Appropriate authorization policies must be set up.
5) Audit must be enabled in order to establish who has had access to the systems.
6) Use of backup and secure copies must be implemented.
7) Controllers of files should write internal regulations and appoint indipendent persons for the security

As regards the conservation of data, it is stated that medical data should be kept no longer than the necessary to achieve the purpose for which they were collected. Moreover, the data subject can usually request data to be erased. Anyway exceptions can be done in the interest of public-health and/or medical science in general. As concerns anonymous data, it can be stored and used without any time limitations as it isn’t protected by the law. This fact has consequences in the treatment of data destined to scientific research. It is mandatory to store data in anonymous way, but some exception can be done if the data subject or his/her legal represantative, or the body designed by domestic laws, must give the consent. In this latter case personal data used for scientific research

should not be published in a form which enables the data subjects to be identified unless under specific consent.

II. State of the art

When dealing with security, it has to be defined which are the assets we want to protect, which are the possible threats, and which can be the solutions. In an information technology context the assets are defined as of three kinds: Hardware, Software and Data.

In the e-Health scenario each asset has a huge importance as long as the system must provide a service with high reliability and guarantee the confidenciality of the data.

For each asset it must be granted [3]:

- Availability
- Authentication
- Integrity
- No repudiation
- Confidentiality

A. Availability

Availability is garuanteed when all the services are available everywhwere (in space) and always (in time). Very often e-Health architectures rely on the public networks, thus the availability in space and time depends on the availability of the network infrastructure.

B. Authentication

Is the process of attempting to verify the identity of the sender of a communication, such as a request to login (usually involves the audit, the record of completed and attempted accesseses).

C. Integrity

Integrity is guaranteed when data can be modified only by authorized users, according to their specific roles.

The last two problems are resolved by what is called ”secure access”.

In this paper two kinds of tools that perform identification, verification and permission assignment tasks are proposed: LDAP and general purpose databases. LDAP (Lightweight Directory Access Protocol) [4] [5], is a distributed information directory that allows people access and permission information storage. It is a common solution when dealing with lots of user accounts, but not strictly needed in an architecture where each service provider has its own independent infrastructure. General purpose databases can be used instead of access control specific technologies, in order to use a single platform for all the data related to users: personal data, administrative data, medical data, login data and so on. The implementation must take into account that the R(97)5 requires that the databases must be separated.

This paper focuses on web services as the technology used for transmitting data between the involved roles of a e-Health infrastructure. Web services are a standard system designed to support interoperable machine-to-machine interaction over a network. Systems interact with each other in a manner prescribed by an interface that is described in a machine-processible format (WSDL), and using messages, which may be enclosed in the SOAP protocol. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards. OASIS [6] and the W3C [7] are the primary committees responsible for the architecture and standardization of web services. When using web services, authentication can be done exploiting the HTTP capabilities for such purpose. Following, the most commonly adopted techniques are shown:

1) HTTP Security: These are the methods that use the HTTP capabilities [8] for authentication. For public web services, it is possible to use Basic, Digest, or a SSL-based scheme.

The Basic process [9] involves transmitting both the user id and password that make up the credentials in clear text to the web server. This information is then validated against the security information maintained by the Operating System. It is easy to implement, requires only one round trip, but carries with it the requirement to encrypt every call since the credentials are sent in essentially clear text. The big advantage of this method is that just about every web service client can work with Basic.

In the Digest scheme [10] passwords are no longer sent clear text, the credentials are hashed on the client so this mechanism does not necessarily require transport encryption. It is a little tougher to implement, but can still be done in a single round trip after the initial authentication.

For really high end security, a very common way of securing web services communications is by means of the HTTPS protocol [11]. HTTPS is based on the SSL protocol which provides peer negotiation for algorithm support, Public key encryption-based key exchange and certficate-based authentication, and symmetric cipher-based traffic encryption [12]. Following this scheme, clients and servers must be configured to use the SSL protocol . SSL, or the later TLS protocols, provide authentication both of the server an the client. This means that during the initial attempt to communicate with a server over a SSL connection, that server will present the client with a set of credentials, in the form of a ”Certificate”, as proof the site is who and what it claims to be. The server may also request a Certificate from the client, asking for proof of the user identity. In order to implement SSL, a server must have an associated certificate for each IP address that accepts secure connections. Commonly, certified X.509v3 [13] certificates are used. Those certificates are cryptographically signed by their owner, and are therefore extremely difficult for anyone else to forge.

For sites in which authentication of identity is important, a Certificate is typically purchased from a well-known Certificate Authority (CA). CAs are responsible for managing certificate requests and issuing certificates, they provide centralized security key and certificate management for the participating devices. Getting the certificate from a well-known CA trustpoint is highly recommended for a secure authentication. However, if a CA is not configured for the routing device

running the server, the server will certify itself and generate the needed key pair. Self-certified certificates are simply user generated certificates which have not been officially registered with any well-known CA, and are therefore not really guaranteed to be authentic at all, anyway they are necessary in order to encrypt the communication. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client will generate a notification that the certificate is self-certified, and the user will have the opportunity to accept or reject the connection. In many cases, however, authentication is not really a concern. An administrator may simply want to ensure that the data being transmitted and received by the server is private and cannot be snooped by anyone who may be eavesdropping on the connection.

SSL can be used both for simple communication cryptography and for authentication. In the first case no Certification Authority is required, the web server only transmits its self-signed certificate to the client in order to encrypt the messages. In the latter case a CA is needed in order to authenticate both the users and the server. The CA does not need to be known worldwide, it can be limited to the Health-Care product scope.

2) SOAP Security: In order to allow end-to-end security, authentication must be placed at the SOAP level by including authenticators in the SOAP headers. There are many standard extensions to the SOAP protocol that provide such functionality. The most adopted is referred as WebService Security (WSS) [14], [15]. The WSS protocol contains specifications on how integrity and confidentiality can be enforced on Web Services messaging and includes details on the use of certificate formats such as X.509.

3) Login method: The idea here is that the client calls a login method on the web service with credentials, and it's passed back a token which is sent with every call, perhaps in a custom SOAP header. It is very similar to the method mentioned above but it requires a minimum of 2 round trips for any single web service call (3 if a logout method exists to clean up the session) and implies a session model on the services, which in many cases map better to a session less model

4) Credentials passing: It is possible to pass the user credentials by simply putting them into the web service call as input paramaters. Is a very simple implementation and can be mapped only to session less models as credentials are passed at each call to the server.

D. No repudiation

No repudiation means ensuring that a contract cannot later be denied by one of the parties involved. The only way to assure no repudiation is having a trusted third party that testifies the contract.

E. Confidentiality

Confidentiality is guaranteed when a sent message cannot be read or modified during the way from sender to destination, being sure that he/she is the person who says to be and preventing that messages could be read or altered in the way.

As already stated before, HTTPS can be easily used for communication encryption. HTTPS has successfully been used in critical security applications as the access to banking details and electronic shopping by Internet, thus is considered a standard de facto for channel confidentiality over the Internet.

Another possible way of encrypting web service messages is by means of Web Service Security. The problem related with HTTP security is that messages remain clear to intermediaries which run counter the SOAP design. If an untrusted proxy was present in the architecture, it would be able to read the messages, thus jeopardizing the security of the system. These methodologies offer security to messages point-to-point and not end-to-end.

In order to resolve this problem WS-Security can be used. WS-Security incorporates security features in the header of a SOAP message, working in the application layer, thus ensuring end-to-end security. The drawback is that it adds a significant overhead, for example needing to encode keys and message signatures into ASCII before sending.

III. TECHNICAL IMPLEMENTATION

This paper shows the results of the research activities within the MyHeart european research project. MyHeart is a 6th Framework Project of the IST Programme, which aims to empower citizens to fight cardio-vascular deseases by means of a preventive lifestyle and an early diagnosis. The scenario we depict comprises three types of actors: patients, medical staff, and service providers. Service providers supply e-Health facilities like remote health status monitoring, access to treatements information, communication with medical staff and psicological support. Basically, they provide monitoring and communication facilities between patients and physicians. In order to accomplish such tasks each patient is provided with some sort of electronic devices with communication capabilities, like a PDA, a mobile phone, or a common personal computer, and measurement capabilities, such as an electronic blood pressure measuring device, an electrocardiograph, or a holter. On the other side, medical staff is provided with a device capable of displaying a graphic user interface of an application (usually a common web browser). In between, there is a set of servers whose task is to store medical data sent by the patients devices, to compute it and to provide additional functionalities such as collaborative environments, messaging, visits and appointments management.

Within the MyHeart project it has been proposed an architecture that comprises:

- A Patient Sation, which collects medical data from the patient, pre-processes and sends it to the servers and acts as an interface to the user.
- A Product Server, which collects all the patients data and post-processes it with specific algorithms that computes a general patient's health status.
- A Portal, which acts as a web interface to the professionals.
- A Professional Device: which implements a common web browser.

Communication within those elements is achieved by means of Web Services protocols.

Such architecture is very general and can be applied to a big number of e-health systems, thus the security issues treated here are also appliable to other similar projects.

In this kind of scenario, we show the possible security threats, which are the legal responsibilities of each actors according to the actual European privacy legislation framework, and a possible techonological solution.

As regards availability, as long as the proposed architecture is provided over the Internet, the problem concerns mostly the communication between the Patient Station and the Product Server. In MyHeart the Patient Station makes use of the GPRS network which is higly available in western countries. The responsibility of making the network available are of the public network owner which should subscribe a Service Level Agreement with the e-Health Service Provider.

As for the authentication and integrity problem (secure access), it must be granted from both the patients side and the professionals side. From the professionals side the access is controlled from the portal. The portal uses a web developing framework called Cocoon [16], which implements a login module capable of retrieving access data both from an LDAP server or a general purpose database.

From the patients side, they are allowed to connect both to the portal and to the product server. The connection to the portal is done in the same way as for the professionals, thus requires the same access control methods.

The connection to the product server is done from the user application. In this case authentication of the patient station can be done exploiting the Web services and HTTP capabilities for such purpose.

In the previous section we proposed two solutions: the basic user-name and password method, which is highly reliable as long as the communcation is encrypted, or by means of a Certificate Authority. Within the MyHeart Project the basic username and password has been developed, anyway the use of a CA is going to be taken into consideratin in the near future. As regards confidentiality https has been used as a standard and raliable tehnology to encrypt communcation.

Legal responsibilities concern both the Service Provider and the users. Users are given a unique username and password and must not communicate them to anybody else. Service Provider, instead, must provide a secure technological solution that prevents malicious users from entering the system without a proper permission.

As regards confidentiality, the MyHeart implementation uses HTTPS as a standard and reliable way for encrypting communication.

Finally, the no repudiation problem has not been taken into account as in the described context one of the parties is the Service provider, which is supposed to be trusted and legally responsible.

IV. Conclusion

This paper proposes a general architecture for e-Health remote patient monitoring, discusses all security problems related to it, and some possible solutions. All the work has been conducted taking into account the European Recommendation No R(97)5 as the legal basis.

The proposed architecture is based on widely known technologies like the web, used as interface to the professionals and web services for patients devices and service provider servers communication. Among these technologies many security solutions are possible. The paper gives a general overview of such solutions and proposes a specific framework based on interoperability and well-known standards.

The overall proposed architecture and its security solutions can constitute an example for the implementation of current future e-Health systems among the european and pan-european research community.

Acknowledgments

The security framework described within this paper is will be implemented in the MyHeart project, Fighting Cardiovascular Diseases by prevention and early diagnosis (IST-2002-507816). MyHeart is a 6th Framework Project of the IST Programme, partly funded by the European Commission.

References

[1] Directive 97/66/ec. [Online]. Available: http://europa.eu.int/eur-lex/pri/en/oj/dat/1998/l_024/l_02419980130en00010008.pdf
[2] Recommendation no r(97)5. [Online]. Available: http://www1.umn.edu/humanrts/instree/coerecr97-5.html
[3] Information security, wikipedia definition. [Online]. Available: http://en.wikipedia.org/wiki/Information_security
[4] J. Hodges and R. Morgan, "Lightweight directory access protocol (v3)," RFC 3377, Sept. 2002. [Online]. Available: http://tools.ietf.org/html/3377
[5] W. Yeong, T. Howes, and S. Kille, "X.500 lightweight directory access protocol," RFC 1777, Mar. 1995. [Online]. Available: http://tools.ietf.org/html/1777
[6] Oasis home page. [Online]. Available: http://www.oasis-open.org/
[7] W3c web services activity. [Online]. Available: http://www.w3.org/2002/ws/
[8] R. Fielding, U. Irvine, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – http/1.1," RFC 2616, June 1999. [Online]. Available: http://tools.ietf.org/html/2616
[9] Basic authentication, wikipedia definition. [Online]. Available: http://en.wikipedia.org/wiki/Basic_authentication_scheme
[10] Digest authentication, wikipedia definition. [Online]. Available: http://en.wikipedia.org/wiki/Digest_access_authentication
[11] Https, wikipedia definition. [Online]. Available: http://en.wikipedia.org/wiki/Https
[12] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol, version 1.1," RFC 4346, Apr. 2006. [Online]. Available: http://tools.ietf.org/html/4346
[13] S. Santesson and R. Housley, "Internet x.509 public key infrastructure authority information," RFC 4325, Dec. 2005. [Online]. Available: http://tools.ietf.org/html/4325
[14] (2006) Web services security, wikipedia definition. [Online]. Available: http://en.wikipedia.org/wiki/Web_Services_Security/
[15] Oasis web services security (wss) tc web page. [Online]. Available: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
[16] The apache cocoon project. [Online]. Available: http://cocoon.apache.org/