Developing Secure Web-based Medical Applications[*][+]

S. Gritzalis [a,b], J. Iliadis [b], D. Gritzalis [c], D. Spinellis[b], S. Katsikas[b]

[a] Department of Informatics, Technological Educational Institute (TEI) of Athens
Ag. Spiridonos St., Aegaleo GR-12243, Greece, e-mail: sgritz@aegean.gr
tel: +30-1-5910974, fax: +30-1-5910975
[b] Department of Information & Communication Systems,University of the Aegean,
Research Unit, 30 Voulgaroktonou St., Athens, GR-11472, Greece
e-mail: (jiliad, dspin, ska)@aegean.gr
[c] Department of Informatics, Athens University of Economics & Business (AUEB)
76 Patission St., Athens GR-10434, Greece, email: dgrit@aueb.gr

Abstract: The EUROMED-ETS pilot system offers a number of security functionalities using off-the-shelf available products, in order to protect Web-based medical applications. The basic concept used by the proposed security architecture is the Trusted Third Party (TTP). A TTP is used in order to generate, distribute and revoke digital certificates to medical practitioners and healthcare organisations that wish communicate securely. Digital certificates and digital signatures are used to provide peer and data origin authentication and access control. The paper demonstrates how TTPs can be used effectively in order to develop medical applications that run securely over the World Wide Web.

Keywords: Trusted Third Parties, Security, Cryptography, Digital Signatures

Acknowledgements: This work was supported in part by the EUROMED-ETS project of the European Commission (DGXIII/INFOSEC/European Trusted Services Programme, EUROMED-ETS project #20820).

1. INTRODUCTION

1.1 The need for supporting modern medical information systems

Computerised information systems allow us to store and handle vast amount of data. The scenery of these systems is rapidly changing because of the massive use of data communications. As a consequence, the number of users that have access to data networks is increasing rapidly. This development has its parallel in the medical sector, where systems are used to store various kinds of patient information and where advanced imaging equipment can directly be coupled to databases that store the images in digitised form.

Figure 1 exemplifies four generic requirements of a healthcare/hospital information system [1]. Such an information system can, nowadays, be used to remotely provide patients with a number of healthcare services. In this case, information and communication technologies are the technological means for the realisation of telemedicine. In other words, telemedicine is the interactive audio-visual communication between healthcare providers and their patients or other healthcare providers regardless of geographical distance.

Several research activities refer to telemedicine. EUROMED is such a project, with the objective to exploit, combine and support high performance computing networking activities, in order to enhance and standardise visualisation techniques to be used in medical applications over Europe. The project utilises the World Wide Web (WWW) as a navigational medium to remotely access multimedia medical information. It is based on interlinked HTML pages which allow

[*] *Medical Informatics*, 24(1):75-90, 1999.

[+] This is a machine-readable rendering of a working paper draft that led to a publication. The publication should always be cited in preference to this draft using the reference in the previous footnote. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

authorised users to access medical data, input them to Java applications invoked from other pages and archive the results by updating links to the old pages.

Three hierarchical infrastructures have been created in the course of the project [2]:

- The Hierarchical Communications Network (HCN); An infrastructure using the Internet, satellites and telecommunications networks (e.g. ISDN, ATM) in order to connect dispersed isolated regions.

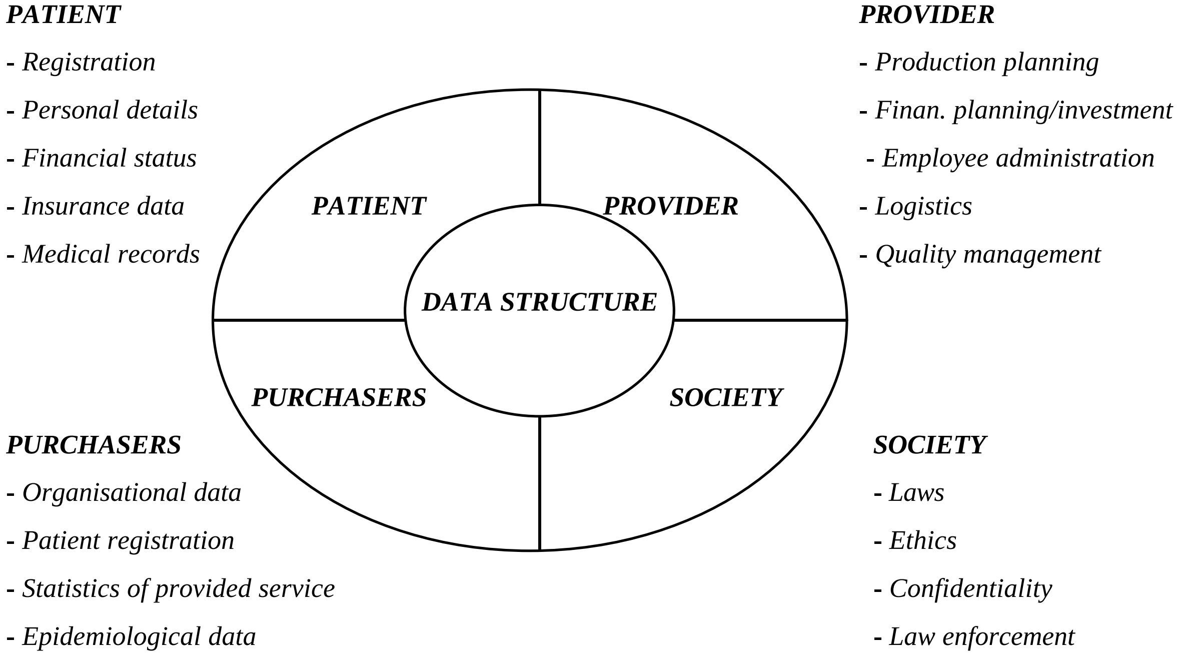

Figure 1. *Baseline requirements of a Hospital Information System*

- The Hierarchical Computing Facilities Infrastructure (HCN); It includes a range of high performance computing platforms powerful workstations and PCs, providing heterogeneous computing facilities to every node in the HCN.
- The Hierarchical Medical Facilities Infrastructure (HMFI); It consists of specialised clinics, general hospitals and local doctors, which can collaborate and facilitate a uniform level of medical practices.

1.2 Web security issues

In recent years, advances in computing and telecommunication technologies have greatly expanded user requirements, applications, functions and tools available to all users of data processing systems, in almost every field of application. Because of the different components, operations, resources and users, computer networks, and especially Internet, are becoming a very convenient target for attacks and illegal operations, a non secure domain. From a general perspective, security refers to a complex of measures, which may be broadly classified as procedural, logical, and physical, and which are aimed at the prevention, detection, indication, and correction of certain kinds of system misuse, both accidental and deliberate [3].

Moreover, the Web aims to provide its user a pool of human knowledge for sharing information and ideas. Web is a distributed hypertext-based information system and includes a body of software and a set of protocols and conventions that allow Internet users to access data through the use of a Graphical User Interface. The Web's hypertext and multimedia techniques make it easy for any user to roam, browse and contribute. From a designer point of view, the Web is based on a client-server model. It consists of a set of servers, known as Web servers, which receive one request at a time and respond to that request without preserving state information, and a set of clients, known as Web browsers, which make requests based on user input and present results. Several efforts have been undertaken to address security in the Web, although the primary focus has been at the application level. These efforts have addressed the issue of protecting the privacy, accuracy and authenticity of transactions conducted over the Internet [4,5].

The rest of this paper is organised as follows: Section 2 refers to the security requirements addressed for medical applications. In section 3, a framework for the description of security services is proposed. Section 4 refers to the implementation of the security framework and functionalities to the healthcare sector, using client/server technologies. The validity of the proposed security services are assessed in Section 5. Finally, in section 6 a number of concluding remarks are outlined and future research directions are briefly discussed.

2. SECURITY REQUIREMENTS FOR MEDICAL APPLICATIONS

2.1 Security Threats

Undoubtedly, any system that is based entirely on the Web for its functions is vulnerable to serious security threats. The most important threats, which we had to deal with in order to establish a secure medical environment, are briefly presented below.

- *Monitoring of communication lines*: By monitoring communication lines wiretappers may gain unauthorised access to medical data, thereby violating the patient's privacy.
- *Shared key guessing*: If one succeeds in guessing a shared key, that specific communication session can be decrypted by him and thus lead to a leak in of a patient’s medical data.
- *Shared key stealing*: If one manages to steal the shared key, that specific communication session can be decrypted by him and thus lead to a leak in of a patient’s medical data.
- *Unauthorised modification of information in transit*: Medical records may be modified on their course to their recipient. Modification may be performed in such a way that the receiving entity will not be aware of the modifications performed.
- *Forged Network Addresses*: If two or more healthcare organisations decide to trust the validity of data that is transmitted from one to the other, then a third party may transmit false medical information that will be accepted as valid by one of these organisations, by forging the network address of the computer that originated the data transmission.
- *Masquerade*: Users may masquerade as valid local or remote users, causing accountability problems. In addition an ingenious intruder may substitute a whole site with a masquerade one, creating thus a weak link in the trust model used by the communicating medical organisations.
- *Password stealing*: When passwords are used to authenticate medical personnel in a network and especially when they are not transmitted in encrypted form (which is usually the case), one may steal these passwords and therefore gain access to the medical resources that are available to the legitimate owner of that password.
- *Unauthorised access*: unauthorised access from invalid users may cause the storage of false, corrupted or modified data, resulting the false diagnosis of a patient.
- *Repudiation of origin*: One may succeed in establishing a communication with a server holding medical data, having successfully forged the communication origin. In case, he will transmit, receive or modify medical records, these actions will be charged to the medical professional or organisation whose address is being forged, during that communication.
- *Private key stealing*: By stealing the private key of an entity, one may succeed in signing digitally an illegally modified medical record or diagnosis and thus validate the new, probably invalid, information contained in that medical record or diagnosis.

- *Private key compromise*: If the private key of an entity is compromised, one may use it in order to sign digitally an illegally modified medical record or diagnosis and thus validate the new, probably invalid, information contained in that medical record or diagnosis.

2.2 User requirements

The EUROMED-ETS project aimed at effectively facing the security threats existing in such an environment. In detail, the requirements which have been proposed by the EUROMED project can be considered as the baseline requirements for any modern distributed medical application and information systems. These requirements are the following [2]:

- The communication infrastructure used is the Internet,
- HTTP or other Web protocols shall be used as the transport mechanisms,
- All security services, which will be implemented for dealing with the threats mentioned before, shall be application transparent,
- Technology proposed must be widespread,
- Proposed solutions must be supported by equipment ranging from a single PC to high performance clusters,
- Sites, regardless of size, power of equipment or location shall hold medical data that need to be made available to other sites in a secure manner,
- A variety of sites must be taken into consideration, ranging from large organisations to isolated medical stations,
- No sites other than TTP sites should be required to take over security functions that will be proposed.

In the context of providing measures like digital signatures in order to prove authenticity and integrity of data, and encryption in order to provide confidentiality, technical, Organisational, medical and ethical aspects have been considered. However, the first level of security was to protect the access to the personal homepage related to every patient.

3. TOWARDS A FRAMEWORK FOR SECURITY SERVICES

3.1 General architecture

Most of the security problems mentioned before can be solved by applying cryptographic methods. There are two general forms of key-based cryptographic algorithms:

- *Symmetric algorithms,* which use the same key to encrypt and decrypt the message. The security of a symmetric algorithm rests in the key and therefore the key needs to be secret.
- *Asymmetric algorithms*, which use a public key to encrypt the message and a private key to decrypt it. The name public key comes from the fact that you can make the encryption key public without compromising the secrecy of the message or the decryption key. The secret key must remain hidden in the owner's domain.

The main advantage offered by public key cryptosystems lies with the fact that it is more scaleable to very large systems, it has more flexible means of authentication, it can support digital signatures and it enables non-repudiation enforcement to verify the transmission or receipt of a given transaction. However, public-key algorithms make the key management process easier, but the need for entities to make their public key widely known poses new problems. At first, new mechanisms to implement publication of these keys could be developed, for the mechanisms commonly used are insecure themselves. Web pages, white pages directories, finger files and Domain Name System are mechanisms commonly used or considered at a first stage. However, it is not possible merely to store a public key in these mechanisms, as the key itself could be modified and the whole process could cause an integrity violation at a user's public key.

The assurance scheme is rapidly improved and can become fully acceptable when it is based on the use of a public-key certificate. It is an information package which includes the user's identity, the user's public key, and it is digitally signed by a trustworthy entity, which is known as the

Trusted Third Party (TTP). TTP is described as "*…an impartial organisation delivering business confidence, through commercial an technical security features, to an electronic transaction. It supplies technically and legally reliable means of carrying out, facilitating, producing independent evidence about and/or arbitrating on an electronic transaction. Its services are provided and underwritten by technical, legal, financial and/or structural means*" [6]. When this scheme is applied to a security infrastructure based on public key techniques, the TTP is widely known as Certification Authority (CA). A CA is made public and certifies that the key is valid for a certain period of time.

When a community of users grows, a single CA may become overloaded because of the big number of certificates it has to manage. Furthermore, each company from the private or public sector, wants to control the way its users generate public keys, and the validity period of the certificates. This causes the creation of various CAs, everyone of which may have a different security policy. This situation introduces an extra problem of Trust Models. In the case of multiple CAs, should the entities that trust different CAs wish to communicate, then the CAs should have organised a way to certify each other, in such a way that there be an acceptable level of trust between them.

There are several possibilities how this type of trust can be realised [7]:

- *No real trust model*: This model is based almost entirely on users mutual exchange of keys prior to the initiation of the first communication. This means that trust exists only in bilateral assurance and it is not applicable for complex organisations. Pretty Good Privacy (PGP) is an example of this model [8].
- *Single trusted arbitrator*: This model is based on the existence of a trusted arbitrator for each transaction. The Kerberos TTP authentication protocol [9] is an example of this type.
- *A set of CAs*: This model is based on CA receiving their trust from a broad user community, due to their commitment to the certification process and public control. Considerable infrastructure is required before users can start making meaningful use of the service. Privacy-Enhanced Mail (PEM) is an example of this model [10].
- *Cross-certification model*: This model consists of a series of CAs. Some of them can cross-certify each other. Such a model was recently implemented by the TESTFIT project [11] whose purpose was to establish a pan-European network of inter-working TTPs providing services for the inter-modal freight transport community.
- *A hierarchy of CAs*: In this model each CA is certified by another CA at a higher level, thus achieving a hierarchy of trust. Each certificate is validated by traversing through the signature chain, verifying each certificate up to the root. Secure Electronic Transaction (SET) is a typical example of this model [12].

3.2 A generic model of TTP services

The major TTP functionalities are briefly presented in the sequel. In Figure 2, the security-relevant functional decomposition at the highest level is demonstrated [13]. The decomposition is focused in the area between the transaction support functions and the TTP functions themselves. Any entity involved in the transaction process makes use of the IT system functions offered for secure transaction support.

Secure transaction support functions in the IT system are decomposed into an Unsecured Support Component and a Secure TTP Infrastructure. Another component composes their services into the required secure transaction support functions. This must itself be secure because it includes at least one interface with the transacting parties, which must not misrepresent the IT encoding of the transaction. But it may handle other security functions all controlled by the respective transacting parties.

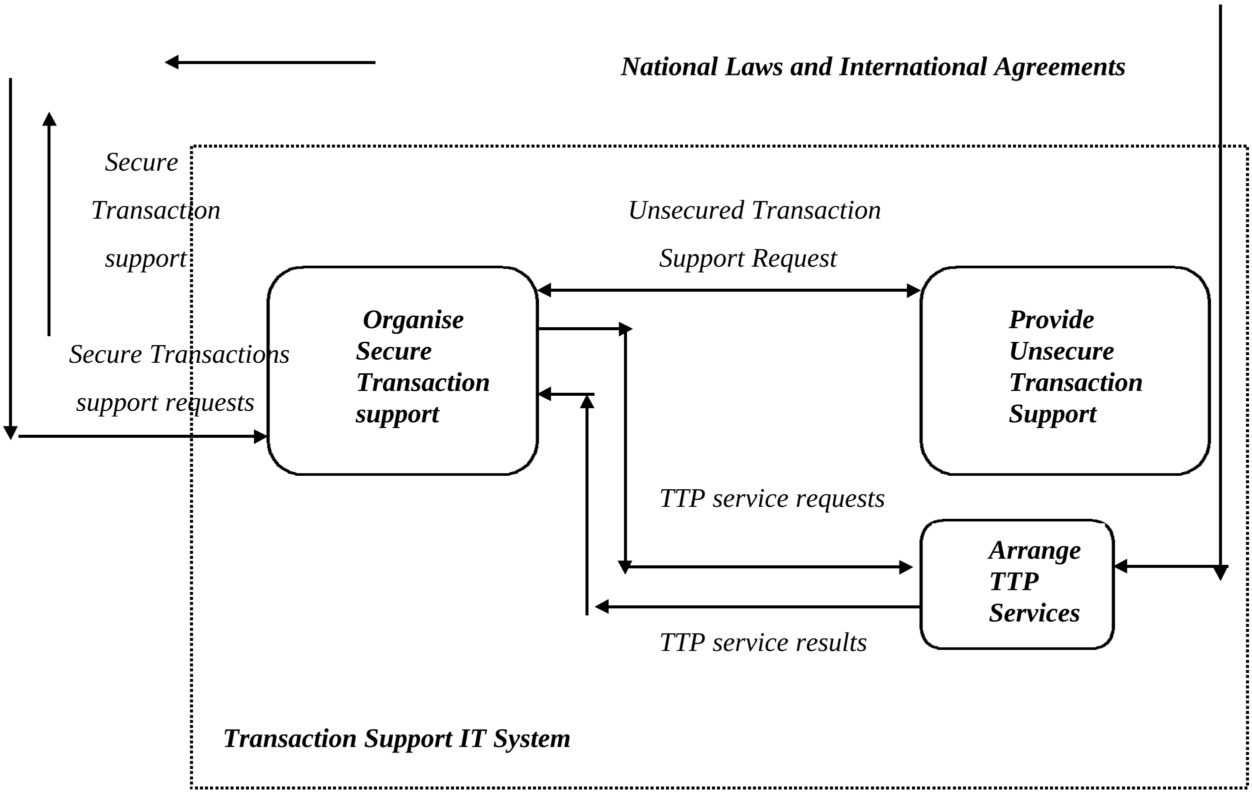
National Laws and International Agreements
Secure Transaction support
Secure Transactions support requests
Unsecured Transaction Support Request
Organise Secure Transaction support
Provide Unsecure Transaction Support
TTP service requests
Arrange TTP Services
TTP service results
Transaction Support IT System

Figure 2. *A generic model for secure transaction support from TTP services*

3.3 TTP functions and software

Electronic Registration

Every user who wishes to communicate with an entity in a specific networking group must register with the appropriate CA. This includes the insertion of the user's identification data in the Directory and the issuance of a certificate for that entity from the CA.

Initialisation

The initialisation of a secure communication session is being established with the exchange of information pertinent to the selected cryptographic algorithms that will be used, the exchange of the authentication keys of the communicating entities and the creation and exchange of a random encryption key, valid only for that session.

Authentication

Authentication comprises the actions that have to be performed in order to verify the identity of an entity. Once this is verified, a certificate is issued for that entity.

Key Personalisation, Generation, and Repository

Key Personalisation is the process of associating a keypair to the registered name of an entity. The keypair is created by the user and the public part of it is communicated to the CA. If the transmitted key does not belong to another entity, then the CA certifies that it belongs to that entity by signing it and thus creating the digital certificate of that entity.

Naming

The naming of the entities is performed in accordance with the X.500 [14] specifications in order to provide the means to identify these entities, without depending on the various identification methods used inside the organisations that these entities belong to.

Certificates: Structure, Generation, Distribution, Storage, and Retrieval

Certificates are signed using the 1024 bit CA's private key. CAs send the issued certificates to the Directory and keep a backup copy at a local repository. The certificate is transferred to the user by means of e-mail; the user may also download the certificate from a specific URL he is informed of, when the certificate is issued.

Auditing

Detailed audit records are kept. Every action performed by one of the modules, the TTP is comprised of, is recorded.

Certificate Directory Management

The Directory acts as a distributed repository of identification and authentication information, such as the user certificates; it is an implementation of the Lightweight Directory Access Protocol (LDAP) protocol [15]. Servers consult the Directory in order to retrieve the latest version of the CRL and identification data for a user that is trying to access them. The communication with the Directory is taking place with LDAP over Secure Socket Layer (SSL) [16].

CRLs: Structure, Generation and Maintenance, Distribution, Storage, and Retrieval

Certificate Revocation Lists (CRL) are lists that contain the certificates that have been revoked. They include information, such as the CRL issuer's identifier, the serial numbers of the revoked certificates and the date each certificate was revoked. The CRL is signed by the CA, using its private key. The CRL is published in the Directory, so everyone may access it.

Date and Time Stamping Services

Security data, such as certificates or CRLs, are always time-stamped. However, this need did not arise for the medical data. Should this be considered necessary in any case, timestamps, signed with the signature of the entity that transmits the data, can be affixed to that data.

4. IMPLEMENTING TTP SERVICES FOR HEALTHCARE OVER THE WEB

4.1 Technical infrastructure

The research work performed in the course of the EUROMED-ETS project (the project aiming at enhancing security functionalities to the EUROMED framework) has led the authors to the conclusion that a framework for securing Web-based medical applications can be developed by establishing Trusted Third Parties and exploiting the services provided by them.

In our case, the target of trust was achieved through a hierarchy of CAs. In this model each CA is certified by another CA at a higher level, thus achieving a hierarchy of trust. Each certificate is validated by traversing through the signature chain, verifying each certificate up to the root. On the other hand, TTPs compose a security scheme which, due to their open architecture and their interoperability with many applications operating on the Web environment, can provide solutions to the security threats that a Web-based medical application may have to confront with. During the pilot implementation of the EUROMED-ETS project, the TTP security scheme was composed of the following modules [2]:

- *Directory Services* acting as repositories of identification and authentication information of the entities participating in the Security Architecture (e.g. users, servers, workstations). The Directory Server that has been chosen is Netscape’s Directory Server v1.02. It supports LDAP v2 and referrals, as they have been introduced to the LDAP protocol [17].
- *Certificate Servers* providing the X.509v3 certificates [18] and thus validating the signatures of the aforementioned entities. Netscape’s Certificate Server v1.01 has been chosen as a CA. It can issue certificates for SSL-based authentication, S/MIME [19] and object signing. It supports all the standards, required from our framework for securing medical applications (e.g. X.509v3, SSLv3, LDAP and PKCS). It provides full LDAP integration; the certificates and other user data may be published directly to a LDAP Directory, besides being stored in a RDBMS.

- *Secure Web Servers* operating as platforms for the execution of the Web-enabled, medical applications. The Web Servers can either host an entire medical application or provide a Web front-end for a standalone medical application, operating at a local level. The Web Server that has been used is Netscape Enterprise v3.0 provides the SSL v3 encryption scheme and LDAP integration, so that the users of the medical application and their respective access rights can be stored directly to a LDAP Directory.

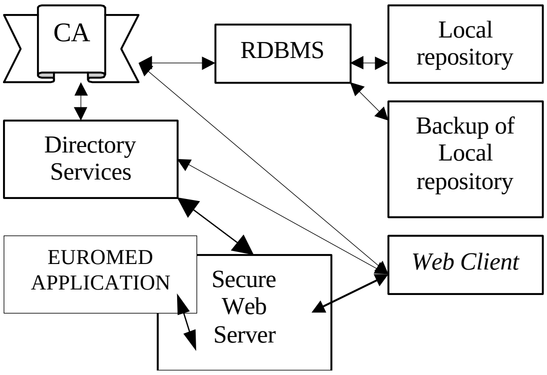

Figure 3. *TTP infrastructure*

4.2 Proposed solutions to security threats

Appropriate measures have been developed to deal with the security threats previously described. Each measure is mapped to the pertinent TTP functions, where applicable, and to actions that need to be taken by the end-entities.

- *Monitoring of communication lines*: Communication between the client and the server is encrypted using shared *session keys*. These keys are valid only for the duration of a *session* (a session is defined as a series of requests and responses between a client and a server).

  End-entities task(s): Encryption using shared session keys (SSL).

  TTP function(s): None.

- *Shared key guessing*: The client generates a pre-master key using a cryptographically secure random key generator. According to SSL, this pre-master key will be the cornerstone for the creation of two session keys and two Message Authentication Code (MAC) keys.

  End-entities task(s): At the client side, the generation of the pre-master key with a cryptographically secure random algorithm (SSL).

  TTP function(s): None.

- *Shared key stealing*: The client encrypts the pre-master key by using asymmetric encryption algorithms. The public key of the recipient entity is used for this purpose by the transmitting end-entity. Key-based authentication is related to the Certification, Key Management and Directory TTP functions.

  End-entities task(s): The client sends the encrypted pre-master key, using the receiver's public RSA key. The receiving entity's public RSA key must be known prior to transmission. That key is part of the receiving entity's certificate information.

  TTP function(s): Certification, Key Management, Key Distribution, Directory.

- *Unauthorised modification of information in transit*: End-entities use secure hashing algorithms for MAC generation.

  End-entities task(s): The messages which are transmitted with MAC hashes, are using the transmitting entity's private RSA key. Receiving entity must verify the authenticity and the integrity of these messages.

  TTP function(s): Certification, Key Management, Key Distribution, Directory.

- *Forged Network Addresses*: End-entities rely on the underlying network, on the TCP/IP protocol implementations at each endpoint and on the Domain Name System (DNS) for this information. Given the open nature of the Internet, these components cannot at present be made totally secure. Thus, end-entities rely on supplementary information (certificates) for the validation of DNS names and network addresses. See *Masquerade* below.

End-entities task(s): outside the scope of EUROMED-ETS activities.

TTP function(s): None.

- *Masquerade*: End-entities must authenticate themselves. Authentication, executed by end-entities, is a separate function for each entity (client, server) and is accomplished by exchange of X.509v3 certificates and by verification of certificate information against the Directory. Authentication closely relates to the Certification, Key Management and Directory TTP functions. Key uniqueness and key personalisation are the required internal TTP functions that render unambiguous authentication possible.

  End-entities task(s): Each entity authenticates itself by sending to the peer a signed message containing its X.509v3 certificate. Entities may choose to verify authenticity of certificates either against a local certificate database or against the Directory.

  TTP function(s): Key generation, Key personalisation, Key uniqueness, Certification, Key Management, Directory.

- *Password stealing*: End-entities restrict password usage: passwords are used, possibly in conjunction with a physical token or other local information, to controlling access to resources local to the end-entities; end-entities never transmit passwords across the network.

  End-entities task(s): Use of password information for access to private key only; avoidance of password transmission.

  TTP function(s): None.

- *Unauthorised access*: Access to resources, usually available through the server entity, is controlled by the entity itself. In order to allow access to a protected resource, the server entity matches the authenticated client identity against a locally managed rule base. This measure is closely related to the authentication process, but is in itself an end-entity function.

  End-entities task(s): Access control, using authenticated peer identity, against a local rule base.

  TTP function(s): None.

- *Repudiation of origin*: End-entities verify, with the aid of the Certification and Directory TTP functions the origin of each network connection. Authentication credentials presented to an entity are logged for subsequent reference. Note that there are two origins, one for the client entity and one for the server entity. The above mentioned TTP functions apply separately to each origin.

  End-entities task(s): Authentication, logging.

  TTP function(s): Certification, Key Management, Directory.

- *Private key stealing*: Keys are protected by both end-entities and TTPs to make as much unlikely as possible their unauthorised use. Key pairs are stored in a human-unreadable format and can be used only when a PIN is supplied. These measures, taken by end-entities, render key-stealing difficult.

  End-entities task(s): Protection, tamperproof-hardware key storage.

  TTP function(s): Key Generation and Initialisation, Key Management.

- *Private key compromise*: Once a key is compromised, any certificate(s) issued with this key need to be revoked. The owner of a compromised key should notify the issuer TTP for any certificates that this key guards. The CRL Maintenance and the Directory functions are used to implement this measure.

  End-entities task(s): Reporting of compromised private keys.

  TTP function(s): Certification, Certificate Revocation, CRL Maintenance, Directory.

From the above, it becomes apparent that not only the TTP, but the end-entity functionality is crucial in the design and implementation of the necessary measures against Web threats.

4.3 A session example

For illustration purposes, an examination of a session example at the technical level is described. Assume that a Physician *P*, using a Client machine *C*, wants to connect to Server *S* of Hospital *H* to obtain the patient record of the hospital Guest *G*. A TTP provides the Certification Authority *CA* and Registration Authority *RA* services.

If *P* is not a registered user in the TTP scheme he must register and obtain a certificate from a *CA*. These steps will allow him to be authenticated by the Server *S*, grant him the rights he may have as a Physician and communicate securely with *S*. *S* must be certified by one of the *CAs* too.

The technical details of the registration and certification process for *P* at the EUROMED-ETS pilot is the following :

[1] *P* will point his browser to the *CA* (if this is the first time *C* is contacting a *CA* to request a certificate, *P* will be prompted by his browser to create his RSA keypair) and apply for a certificate. *P* will be instructed by the *CA* to provide certain identification information.

[2] The *CA* Administrators will forward his request to the *RA* Administrators; they will verify the identification information *P* has provided the *CA* with.

[3] *P* will be asked by the *RA* Administrators to provide any needed additional information to prove his identity according to the *CA* policy.

[4] If his identity is confirmed, an entry for him will be created in the Directory and the *CA* Administrators will be notified.

[5] The *CA* Administrators will notify *P* to download his certificate from a specific URL.

[6] *P* must download and install (automatically) the certificate in his browser.

[7] *P* is now ready to commence secure, authenticated communication sessions with Server *S* of Hospital *H* and any other server that performs lookups in the Directory in order to authenticate the users that request access. If *S* does not possess a certificate, the system administrators of Hospital *H* must follow the same procedure to register and obtain a certificate for *S*. Assume that *P* attempts access to *S*, in order to obtain the patient record of *G*. The procedure, in detail, is the following :

[8] *P* points his browser in client machine *C*, to *S*.

[9] *SSL handshake* occurs and secure communication (SSL) commences between the two parties as soon as *P* enters his secret key password in order to unlock it.

[10] *S* performs a secure lookup in the *CA* Directory. If *P* is a registered physician and his certificate has not been revoked he will be in the "Physicians" group, with a valid certificate. The *CA* Directory performs a search and if the requested data do not exist locally, refers the request to the appropriate Directory branch. When that is reached group membership, certificate and other identification data of *P* are returned to *S*.

[11] *S* compares the certificate presented to him by *C* and the certificate contained for *P* in the *CA* Directory. If they match, authentication was successful.

[12] *S* uses the identification data received from the *CA* Directory to perform a lookup in the local Access Control List. According to the local *ACL*, all physicians have the right to obtain patient records, so *S* grants the right to *C* to access the patient record of *G*.

[13] The patient record of *G* is presented to *P*.

The above procedure is transparent to *P*. He only knows that he has directed his browser to S and requested for the patient record of G, which he has taken.

The exact steps of the *SSL handshake* (step 9), where both parties (Server *S* and Physician *P*) possess valid certificates and have to be authenticated by each other in order to commence secure communication on application level. The SSL handshake protocol is responsible for selecting the ciphers that will be used, the authentication of the client and server and for the creation and secure submission of the pre-master key which will be used for the creation of two session keys and two MAC keys that will be used. Since HTTP operates over the SSL protocol, every HTTP transfer will take place on a higher level than the one we will describe here and thus will be secure.

[9.1] *P* will initiate the communication by pointing his browser to *S*.
[9.2] *C* sends a *ClientHello* message to *S*.
[9.3] If *S* does not respond with a *ServerHello* message the connection will fail.
[9.4] If *S* replies, security parameters such as protocol version, session ID, cipher suite and compression method are exchanged.
[9.5] *S* sends its certificate to *C* and waits for authentication.
[9.6] Once *S* is authenticated (the *CA* of *S* is checked against the browser's list of trusted *CAs*), *S* requests a certificate from *C*.
[9.7] *S* sends to *C* the *HelloDone* message to indicate that this phase is complete.
[9.8] *C* communicates the certificate of *P* to *S*.
[9.9] *C* sends to *S* the *KeyExchange* message.
[9.10] A *ChangeCipherSpec* message is sent by *C*.
[9.11] *C* copies the pending *CipherSpec* into the current *CipherSpec*.
[9.12] *C* sends the *Finished* message under the new algorithms, keys, and secrets.
[9.13] *S* will send its own *ChangeCipherSpec* message, transfer the pending to the current *CipherSpec*, and send its *Finished* message under the new *CipherSpec*.

At this point, the handshake is complete and the C and S may begin to exchange securely application layer data. The two symmetric keys that will be used for the bulk encryption have been created from the server and the client using the pre-master key, according to the SSL process. Furthermore, the pre-master key has been used to create the two MAC keys.

4.4 Legal and Organisational issues

The cornerstone of the security scheme presented in this paper is the use of digital signatures. The legal recognition of the digital signature concept is now emerging in a number of European Union Member States [20], as well as in other countries (e.g. US, Canada, etc.) and it is expected to be completed in the next few years. Furthermore, the medical data that is transferred through the Web is protected by the European Convention on Human Rights, [21] and by the Recommendation on the Protection of Medical Data [22]. In addition, and in the case of European union, TTPs functionalities and operation must meet the requirements set for by the EU Data Protection Directive.

5. ASSESSING THE VALIDITY OF THE PROPOSED SECURITY SERVICES

Technical assessment

TTP services and functions have been implemented and provided. The public and private keys used in authentication were RSA keys of 1024 bits (CA signing key) and 512 bits for the Server and user keys. The encryption algorithm used was RC4/40 [23]. The MAC algorithm used was MD5 [24]. Speed loss in transferring data due to encryption is inevitable. However, the level of speed loss that has been observed was not obstructing the provision of TTP services. Should the volume of transactions provoke an increase in the level of speed loss, there are several alternatives to be considered (e.g. hardware-based encryption, server load-balancing).

The security scheme presented in this paper is based on open specifications (HTTP, LDAP, SSL v3, X.509v3, PKCS7) and therefore provides interoperability with most of the clients and servers operating in the Web-environment. Moreover, for the software that was used (Web Servers, Directory Servers, Certificate Servers), interoperable versions exist for a variety of operating systems, such as Microsoft Windows NT, Solaris and Linux.

Organisational

The infrastructure required from a client in order to access medical applications securely is merely a Web browser, an Internet connection and a registration to the Certificate Authority. The internal organisation of the Directory is a matter that should concern the implementors of each different medical application. It should be noted that human interference is required in certain steps of the operation of a TTP, such as the issuance or the revocation of a certificate, before it's expiration.

Operational

The quality of services provided by a TTP depends on the existence of a Help Desk and the implementation and use of standard procedures for the TTP operation. Administrators were available at any time in order to provide assistance in using the services provided by the TTP. Another factor that contributed to the quality of services provided was the availability of documentation to the end-users and the automation of several procedures pertinent to the services these users were requesting. The distributed nature of the Directory, which was used as a repository for identification and authentication information, rendered this TTP Security Architecture expandable. The addition of new users, Servers or administrators in the Security Architecture has proven to be an easy task that can be partially automated. In order to exploit the security services of a TTP, implementing them would not be enough; the medical personnel that will use these services has to be convinced of their necessity and eager to familiarise themselves with using them.

Financial

A TTP site has to invest on hardware equipment (Servers), software modules for the TTP and an Internet connection through a leased line at least of 512 Kbps. The end-users need only their standard computer equipment, plus a dialup or leased line connection to the Internet and a Web browser. The medical organisations that will deploy healthcare applications and provide related services to healthcare professionals and patients should also be equipped with the necessary hardware (Server), software modules (Web Server and medical application) and an Internet connection through a leased line of at least 512 Kbps. As far as the TTPs and the medical sites are concerned, the cost of upgrading the hardware equipment, when needed, and the cost of training their personnel in the use and administration of the TTP, should be also considered.

6. CONCLUDING REMARKS AND RECOMMENDATIONS

Trusted Third Party services based on SSL provide a viable solution for deploying secure medical applications over the Web, allowing patients and healthcare providers to communicate securely. Should a healthcare institution decide to adopt the TTP solution as a security service provider, wide-scale tests should be performed before the deployment of medical services. The objective of these tests must be to identify and provide solutions for problems that may arise in a wide-scale deployment of healthcare services. Problems that may emerge include the organisation of the security provisions and the definition and implementation of access control lists, depending on the needs of the medical organisations, healthcare professionals and patients that will use the aforementioned medical services.

The United States export policy restricts export of cryptography technologies to countries outside the United States. This obstacle is currently being surpassed as encryption modules, delivering strong encryption, are already emerging in the European market. The open architecture that characterises Trusted Third Party solutions renders them capable of incorporating these new modules, and thus augment the level of security they provide.

The proposed TTP-based security architecture provides the means to secure transactions over the Web. However, one should consider the fact that the security scheme applied in this case is capable of securing other Internet transactions as well, such as Java-based transactions, e-mail exchange, telnet and ftp sessions. The Java security scheme is evolving in order to take advantage of the security provisions of digital signatures. The use of TTPs and digital signatures has already been incorporated in the security scheme applied in e-mail transactions through S/MIME and research is currently performed for securing the telnet and ftp sessions by exploiting the services provided by TTPs and digital signatures.

In conclusion, TTPs can be considered as security solution carriers for most of the Internet services, and particularly for Web-based medical applications. This potential of the TTPs may also provide security solutions in other communication platforms, such as in X.400.

REFERENCES

[1] Smeets, B., Johansson, T. (1997) Secure Storage and Retrieval in Medical Information Systems (Lund: Lund University Press)

[2] Spinellis, D., Gritzalis, S., Iliadis, J., Gritzalis, D., Katsikas, S., et al. (1997) Trusted Third Party Services for Healthcare in Europe, DGXIII/INFOSEC Project 20820 EUROMED-ETS final deliverable

[3] International Standards Organisation ISO/TC97 7498-2 (1988) Information Processing Systems - OSI Reference Manual Part 2: Security Architecture (Geneva: ISO)

[4] Gritzalis, S., Spinellis, D. (1997) Addressing Threats and Security Issues in World Wide Web Technology, in Proceedings of the 3rd IFIP International Conference on Communications and Multimedia Security, (S.Katsikas Ed.) 33-46 (London: Chapman & Hall)

[5] Meyer, K., Schaeffer, S., Baker, D. (1995) Addressing Threats in World Wide Web Technology, in Proceedings of the 11th IEEE Annual Computer Security Applications Conference, 123-132 (Los Alamitos: IEEE)

[6] Brown, A., Gray, G., Lambert, O., Muller, P., Castell, S., Balouet, V., (1993) User Requirements for TTP services, DGXIII/INFOSEC Project S2101 S01 deliverable

[7] Rensburg, A., Solms, B. (1997) A Comparison of Schemes for Certification Authorities, Proceedings of the IFIP SEC'97 International Information Security Conference, 222-240 (London: Chapman & Hall)

[8] Zimmermann, P. (1995) PGP Source Code and Internals (Cambridge: MIT Press)

[9] Millen, J., Neuman, C., Schiller, J., Saltzer, J. (1987) Kerberos Authentication and Authorization system, Project Athena Technical Plan, Section E.2.1 (MA: MIT Press)

[10] Kent, S. (1993) Privacy Enhancement for Internet Electronic Mail: Part 2: Certificate Based Key Management, Request For Comments RFC 1422

[11] TESTFIT - TTP & Electronic Signature Trial for Inter-modal Transport (1995) DGXIII/INFOSEC Project S2303, Final deliverable

[12] MasterCard, Visa (1996) Secure Electronic Transaction Specification, Book1: Business Description

[13] Muller, P. (1993) Functional Model of Trusted Third Party Services, DGXIII/INFOSEC Project S2101 S03 deliverable

[14] CCITT (1988) Recommendations X.500-X.521, Data Communication Networks Directory (Geneva: CCITT)

[15] Yeong, W., Howes, T., Kille, S. (1995) Lightweight Directory Access Protocol, University of Michigan, ISODE Consortium, Request For Comments RFC 1777

[16] Freier, A., Karlton, P., Kocher, P. (1996) http://home.netscape.com/newsref/std/SSL.html

[17] Michigan University Research Team (1997) http://Web.umich.edu/~dirsvcs/ldap/

[18] CCITT Blue Book (1988), Recommendation X.509 and ISO 9594-8, Information Processing Systems - OSI - The Directory Authentication Framework (Geneva: CCITT)

[19] RSA Data Security Inc. (1995) S/MIME Implementation Guide, Interoperability Profile, Ver.1. (Massachusetts: RSA Inc.)

[20] European Commission COM(97)503 (1997) Ensuring Security and Trust in Electronic Communication: Towards a European Framework for Digital Signatures and Encryption (Brussels: DG XIII)

[21] Council of Europe (1981) Convention for the Protection of individuals with regard to automatic processing of personal data, Convention No. 108 (Strasbourg: CoE)

[22] Council of Europe Recommendation R(97)5 (1997) On the Protection of Medical Data (Strasbourg: CoE)

[23] Rivest, R. (1992) The RC4 Encryption Algorithm, RSA Data Security, Inc. (Massachusetts: RSA Inc.)

[24] Rivest, R., Dusse, S. (1992) The MD5 Message-Digest Algorithm, Request For Comments RFC 1321