EUROMED-JAVA: Trusted Third Party Services for Securing Medical Java Applets

Angelos Varvitsiotis, Despina Polemi*, and Andy Marsh**

Institute of Communications and Computer Systems (ICCS)
National Technical University of Athens (NTUA)
Heroon Polytechniou 9, Zografou
Athens, Greece

The authors can be reached via e-mail at the following addresses:
a.p.varvitsiotis@iccs.ntua.gr
polemi@softlab.ece.ntua.gr
andy@naxos.esd.ntua.gr

Abstract. EUROMED, a DG III project[1], aims to create the foundation of telemedical information society. EUROMED-ETS, an INFOSEC project, provided secure communications among EUROMED participants by establishing Trusted Third Party Services (TTPs) over the Web. Java technology plays an important role in EUROMED. In this paper, the threats that Java technology introduces to EUROMED are explored and security countermeasures are proposed, utilizing the TTP infrastructure.
Keywords: Telemedical Applications, Web, Trusted Third Party Services, Java, EUROMED, EUROMED-ETS.

1 Introduction

EUROMED is a three-year European Commission DG III/B project [13][14] [22], which began in January 1996. ICCS-NTUA is leading the project and the partners are: Infoproject (GR), University of Calabria (I), University of Athens (GR), AIS (IT), University of Joensuu (SF), University of Amsterdam (NL), University Hospital Leiden (NL) and University "PUB" Bucharest (RO). EUROMED aims to create the foundation of a telemedical information society. Patients' health records will be distributed in medical cyberspace, and doctors will be able to interact with these records by clicking on HTML pages. EUROMED supports three hierarchical infrastructures, namely:

* Project Manager of EUROMED-ETS
** Project Manager of EUROMED
[1] This paper represents the views of the authors and not those of the European Commission.

- The HCN (Hierarchical Communications Network) using satellite, Internet and telecommunications networks (e.g. ISDN, ATM), which connects dispersed isolated regions.
- The HCF (Hierarchical Computing Facilities) infrastructure, which includes a range of High-Performance Computing (HPC) platforms, modern workstations and PCs, provides heterogeneous computing facilities to every node in the Hierarchical Communication Network.
- The HMF (Hierarchical Medical Facilities) infrastructure, which consists of specialized clinics, general hospitals and local doctors that can collaborate and facilitate a uniform level of medical practices.

EUROMED is based on WWW to establish communication between the participating sites. Its network consists of a number of Internet sites that stores medical data about a number of patients, as well as image processing and archive applications. A physician seeking information to reach diagnosis for a given patient, searches the network with a web browser and collects the available data for the specific patient. Medical data is assumed to be images, two- or three-dimensional, coming from different modalities (X-rays, mammography, CT, MR), biosignals – ECGs–, results of biochemical examinations and text reports. The available data for any given patient does not necessarily reside on one site; instead, each data item is archived on the site where it was first produced. The data is annotated with pointers that refer to one another and are summarized on an HTML page, unique to each patient.

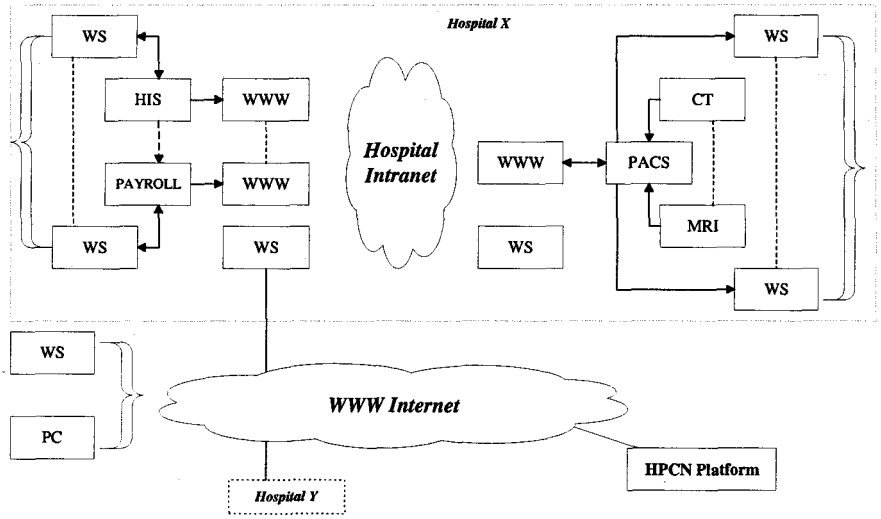

Fig. 1. EUROMED platform

The physician inputs the data of the referred patient into the appropriate application and processes it. An application is any program that is distributed to the EUROMED sites, either on physical media, or over the Internet. Java [10][11] was chosen by EUROMED as the standard technology for the distribution of common applications over the Internet.

EUROMED-ETS is an INFOSEC project [23] which complements EUROMED by concentrating on, and tackling, several issues of security in a telemedical information society [2][3][4]. ICCS-NTUA was leading the project and the partners were: University of Aegean (GR), University Hospital of Magdeburg (D), University of Calabria (I), Infoproject (GR) and National Health Service (U.K.).

In EUROMED-ETS, we concentrated on the establishment of TTPs [6][5], for ensuring that all health actors can communicate in a secure way over the WWW. TTPs secure the highly sensitive medical information exchange. In this project all aspects (operational, technical, functional, organizational, regulatory, and legal) of TTPs have been investigated for telemedical applications over the WWW. SSL [19] was adopted as the security solution after considering EUROMED's needs. Among the services offered by the established TTPs were: registration, name authentication, certification, key and directory management. During the pilot phase of the project a network of four TTPs was established. EUROMED provided the pilot platform in order to demonstrate the proposed security solution.

EUROMED-ETS concentrated on the first and last hierarchy (HCN and HMF) of the EUROMED platform in order to ensure the secure communication of EUROMED users over the Internet. For the proper operation of EUROMED, a complete security solution for all three hierarchies must be provided. Only then EUROMED will reach its true potential and all its participants (among them, 80 Hospitals throughout Europe) will start utilizing its true benefits. In order to achieve this goal we need to provide measures for securing the medical Java applets distributed over the EUROMED network.

In this paper we examine the role of Java technology in EUROMED, we investigate the threats that Java technology brings to EUROMED, we explore the impact of TTPs in improving Java security and we propose security countermeasures based on the TTP technology. This paper is organized as follows: Section 2 concentrates on the role of Java technology in EUROMED and addresses the threats that this technology brings to EUROMED. Available security measures are reviewed and security problems that remain unresolved are described. Section 3 provides countermeasures for overcoming the threats that Java puts to EUROMED's security. Conclusions and topics for further research are included in Section 4.

2 Java in EUROMED – Security Issues

EUROMED has standardized the use of the WWW for telemedical applications. It has proposed HTML, VRML, JavaScript and Java as the standards on which different sites can communicate and interact in order to materialize the

concept of Telemedicine, remote diagnosis and the building of a virtual environment in which physicians interact with each other and with patients' data. Medical data, scattered in different sites even in the case of the same patient, are processed by Java applications, residing in different Institutions and Hospitals. EUROMED even offers the possibility for Hospitals to use HPC installations for CPU-intensive applications. EUROMED is based on a network of HTML pages that allows users to access medical data, input them to Java applications, invoked from other HTML pages, and archive the results by updating links to the old HTML pages.
Java is selected as a key language in EUROMED for the development of medical applications because of its suitability for building Web applications and its platform- and OS-independence properties. The characteristic of Java to generate platform-independent byte code is crucial for EUROMED since different Hospitals, Institutions and Research Centers that participate in the network are bound to use different computer systems, ranging from PCs, to workstations and HPC systems.
Java is used to create applets with medical information that are embedded in HTML pages. EUROMED Java applets perform operations like: image display, image segmentation, multimodal image registration, fusion and anatomical object visualization. A simple mouse click on the browser invokes the execution of the applet on the physician's computer. Summarizing, Java was chosen by EUROMED as the key technology for a variety of functions, including:

- visualization of medical data
- annotation of visual data (e.g., medical images)
- remote database access
- update of medical records
- cooperation with services available by HPC centers for CPU-intensive applications
- assistance in reaching diagnosis.

Applications such as the above are built either as Java applets that are downloaded and executed by the user's browser, or as Java applications, that need to access resources of the user's computer and operating system in order to perform certain operations. We concentrate on the applets case and the security threats that the model of Java applets introduces in EUROMED. In order to discuss these threats, we categorize them as follows:

- *downloading security threats:* this category includes the security threats involved in downloading an applet or any mobile code application [7][12]. The site hosting the applet may be impersonated [1] by another, malicious site. Alternatively, the executable byte code may be altered, either as stored on the hosting site, or during transmission across the network. Maliciously written applets, once loaded instead of legitimate ones, may steal or alter precious information, orchestrate denial-of-service attacks against the hosting server, or consume resources of the client's computer. In EUROMED, an additional severe threat is that malicious applets may cause erroneous diagnoses by presenting false data to the physician.

- *execution security threats:* this category includes the threats that the execution of an applet may involve against the user's computer and the local resources [15]. Typically, a malicious applet may attempt to read or write a file in the local disk, reveal sensitive information about local resources to unauthorized parties, or exhaust local resources such as CPU, memory and network bandwidth.
- *communication security threats:* EUROMED applets need to communicate with servers over the Internet to read and update medical information. The confidentiality and the integrity of stored information and data in transit [1] are threatened if not appropriately secured.

For each of the abovementioned threats, countermeasures have been proposed by the designers of Java [17][18]. In the sequel, we shall assume that HTTP communication between a browser and a server can be protected by authentication for both the server and the client, encryption and access control, as supported by widely-accepted methods and protocols (e.g., SSL [19], SHTTP [20]). Therefore, as far as the applet downloading threats are concerned, the following countermeasures have been proposed:

- The server that hosts an applet must be digitally certified so that no other server can impersonate as the legitimate one.
- The server can request user authentication, so that only authenticated users are able to download applets.
- The server can implement access control, so that only authorized users are able to download applets.
- The communication between the browser and the server must be encrypted using strong encryption [26][27] and authenticated using an appropriate MAC algorithm [21][24][25], so that the confidentiality and integrity of the applet code can be guaranteed to a satisfactory degree.
- The code of the applet itself must be digitally signed by the applet's author, so that the user can avoid execution of applets provided by untrusted sources.

To add a level of trust to the code itself, the designers of Java have taken some measures against execution threats. When loaded by browsers, Java applets execute in a confined environment, as controlled by a Security Manager. The security manager will only allow a restricted subset of operations normally available to native Java programs. Such restrictions are: applets cannot open, read, rename or write system files, cannot examine most of the system- and user-related variables available to programs and cannot establish network connections to any other but their hosting server.
This is too restrictive for EUROMED where reading medical data, processing them, acquiring new data and saving the results either locally or remotely, is essential. These operations presuppose that the Security Manager does not apply the abovementioned restrictions so that applets can perform useful operations (e.g. access and manipulate local files or establish arbitrary network connections). For overcoming this problem, these restrictions should be uplifted in well-defined cases. This will offer to appropriately certified applets the necessary

flexibility in order to perform efficiently by e.g. storing intermediate computation results on local temporary files, at the same time offering trust to the compliance of these applets to issues and directives regarding privacy of medical information and personal data [30][31][33][4][34].
Furthermore, execution threats cannot be entirely handled by means of the above restrictions. Several problems regarding the execution of applets remain unresolved, including:

- *code modularity:* in the event of a complex, multi-component application where trusted code relies on some untrusted component, the security of the overall application is questionable.
- *multithreading:* the freedom given to applets to fork large numbers of threads as well as the possibility to alter the default thread behavior have often been the causes for resource starvation and denial of service problems.
- *implementation opaqueness:* although specifications for VMs and foundations of basic classes are open, their implementations in browsers are not, leaving space for possible security breaches through poor design or implementation choices.

As far as the *communications threats* are concerned, the measures proposed are based on recently announced and made available Java Application Interfaces, such as a foundation of classes called the Java Cryptography Enhancement toolkit (JCE) and SSL, which is now part of the basic Java Development Toolkit (JDK) [18]. JCE contains implementations of widely accepted digital signature algorithms such as DSA and RSA, encryption algorithms, such as DES [26] and RC4 [28], and message authentication algorithms, such as MD5 [21]. It also contains classes for parsing X.509v1 certificates, and the X.509v3 functionality is becoming available. With the use of the JCE classes, Java applets can:

- Establish network connections with one- or two-side authentication, by verifying certificates presented by the other end of the connection.
- Transmit messages using encryption and appropriate MAC codes to guarantee communications confidentiality and integrity.
- Enable access control based on user authentication for critical tasks (e.g., some users may be allowed to view medical information without the right to alter it).
- Provide other functions, such as patient anonymity, where required by the context or the medical application.

Using the SSL API, applets can have secure access to a wealth of applications in a standardized manner. At the same time, SSL provides all of the above functions in the context of a network session, which is particularly useful for the applications in hand (e.g., interactions with medical databases and use of HPC facilities over the Internet). However, there are still problems that need to be resolved, including:

- The degree to which signed applets should be given unlimited access to local resources.

- The lack of a trust chain between trusting an applet author's intentions and trusting the applet code itself.
- The poor security features of the confined execution environment regarding system resource management (number of threads, alteration of default thread termination behavior).
- The possibly excessive strictness of the confined environment.
- The lack of a mechanism for user-configurable, fine-grain tuning of allowed applet access to system resources.
- The lack of a taxonomy for the certification of code other than the name spaces provided by manufacturers.
- The opaqueness of the classes provided by browser manufacturers.
- The premature state or lack of components such as Directory lookup modules [35].

In EUROMED-ETS, we demonstrated the usefulness of TTPs in providing a source of trust for the complex, multi-server, multi-hierarchy, multinational, multi-user, geographically dispersed infrastructures employed by EUROMED. In the next section, we discuss how TTPs can be used for solving the abovementioned problems and address the security threats imposed by the EUROMED applets.

3 TTPs for Securing EUROMED Applets

The establishment of Trusted Third Party Services (TTPs) was proposed and demonstrated by pilot operation during EUROMED-ETS in order to secure HTTP-based transactions. Several security services were provided by EUROMED-ETS in order to establish secure communications among EUROMED participants. However, EUROMED-ETS did not provide countermeasures for the abovementioned threats brought by the use of Java technology. In this section, we propose additional mechanisms to be offered by TTPs and manufacturers in order to secure the medical Java applets distributed over the EUROMED network.

In order to ensure the authenticity and integrity of applets we propose extending the Directory service of TTPs to support directory services applet certification. Execution threats can be overcome with source code certification and varying levels of trust. Secure data transport among applets and servers can be provided either by secure applet communications or by proxying services offered by TTPs. Access to servers that are contacted by applets is secured with access control. Finally, privacy can be enhanced with one-time keys for anonymous services. These mechanisms are explained in the sequel.

Directory Services for Applet Certification: Java classes provided by manufacturers are normally signed by the manufacturers themselves using their private key. During EUROMED-ETS, we used the Directory as a publicly accessible trusted infrastructure for certificate lookup via standardized access protocols [36]. In order to offer a similar service for the provision of a unique source of

trust to applet certificates, the functionality of the EUROMED-ETS Directory should be extended to hold certificates of trusted applet manufacturers. This will enable users to install certificates from a trusted source, using a secure connection (LDAP over SSL) for the Certificate lookup. Besides certificates, the Directory can offer a storage space for Java objects [35]. This is a valid alternative that can also be considered by EUROMED TTPs.

Source Code Certification: TTPs can offer a more important service than just storing manufacturer certificates. If TTPs are allowed, on a confidential basis, to examine the source code of an applet, they will be able to certify it themselves as conforming to specifications set forth by EUROMED. Wherever possible, code should be examined with regard to execution properties. Examples of simple properties are: *all network connections initiated will use encryption; the default thread termination methods will not be altered; the number of threats forked will be bounded;* and so forth. More subtle properties can require more careful inspection and/or object classes specially crafted to this end. Examples of such properties include: *sensitive user data will stored, if need be, appropriately encrypted; network connections will only be initiated to addresses learnt through the hosting server;* and others. Once satisfying a predefined set of properties to the best of the TTP site personnel's knowledge, the applet code can be signed by the TTP. If multiple levels of trust are used as proposed below, the level of trust will correspond to the specific properties that are asserted.

Varying Levels of Trust: Browsers must be updated in order to allow a user-configurable set of operations to each applet. This involves having the applet Security Manager maintain and consult a list indexed, for example, by the known certificates of code signers. For each certificate, the list must contain the set of allowed sensitive operations. Using this capability in conjunction with Source Code Certification as discussed above, applets can be signed with different keys, depending on the operations that they need in order to operate correctly and/or efficiently. Varying levels of trust will be supported, using a different key for each level. Levels will range from "completely untrusted", prevented from execution, to "fully trusted" which can be given full access to local resources. Intermediate levels may be defined e.g., to allow network connections to arbitrary addresses learnt through an existing secure connection, to allow access to a limited portion of the local file system, etc. In this paradigm, TTPs will be responsible for the certification of applets, by signing them with the key for the appropriate security level.

Secure Applet Communications: The certification service of TTPs should be extended to cover the needs of secure communication among applets and available services (e.g. database servers, image servers, etc.). Such connections should base their communications security on a standardized underlying mechanism, and recent Java developments show that SSL is the mechanism of choice.

Thus, certificates should be issued by TTP sites and be installed on the sites that host services that applets use.

Access Control Management: Communication between applets and servers should be two-way authenticated and access-controlled. Access control should be based on the user certificate presented by the client applet and, as suggested in EUROMED-ETS, it should be organized on a group membership basis to allow flexible access control policies and maximum manageability. This requirement has two interesting implications, one for each communicating side. Servers, one the one hand, must be able to verify a large number of certificates, possibly signed by various authorities, as well as to check the Certificate Revocation Lists provided by these authorities. The suggested solution for this problem in EUROMED-ETS was to consult the Directory via a secure connection (LDAP over SSL) to verify the validity of user's certificates as well as to check for the latest version of the CRL. This solution should be generalized to cover servers such as the database and HPC servers needed by EUROMED applications. The second implication is that applets, in order to present the user's certificate to a server, should have the means to access and use the user's private key, usually stored on a smart card or on magnetic media. For this reason, an exception should be crafted in the SSL component of Java so that applets are allowed to read key data from the local device holding the user's key pair. Alternatively, the browsers should provide the required functionality to either import certificates stored in standardized formats [29] on magnetic media, or support access to tokens through the JavaCard API [18].

One-time keys for anonymous services: Due to the confidential nature of EUROMED, users may sometimes be granted access to medical data such as an X-ray image, without any other access (e.g., without access to the patient's medical record and other personal data). A valid scenario would be to have an external physician examine and annotate an X-ray image of a patient and store back the annotated image. In order to allow such applications to interact securely with servers, one-time keys should be used. One-time keys may be issued by TTPs on request from both the server and the user applet wishing to provide to one another one-time access to a resource for read or update purposes. Secure connections should be used for transferring components of one-time keys.

Other Communication services: If the restriction that applets can only communicate with the hosting server is not uplifted, TTPs can serve as communication intermediates among client applets and various services. If TTP sites host applets that need to communicate to several servers to complete a task (e.g., to an HPC site for a computation-intensive task and to one or more database servers for data retrieval and archiving), a TTP site may offer transparent connection redirection or proxy services.

4 Conclusions and Further Research

The use of Java technology in EUROMED introduces new security problems. The extension of the TTP services offered by EUROMED-ETS can provide effective solutions to these problems. This will allow Java applets to be transported securely to the physicians' workstations and executed in a trustful environment that can be fine-grain controlled and configured. The interaction of the proposed measures with emerging technologies, such as biometrics- or Java-enabled smart cards should be explored, in order to enhance access control, authentication mechanisms and privacy in the Healthcare sector.

References

[1] Ahuja, V.: Network & Internet Security. Academic Press, NY 1996.
[2] Barber, B., Bakker A.R. and S. Bengtsson (eds.): *Caring for Health Information: Safety, Security and Secrecy.* Amsterdam: Elsevier Science, 1994.
[3] Blobel, B.: "Towards Security in Medical Telematics: Legal and Technical Aspects," *Open Information Systems and Data Security in Medicine.* Barber B., Treacher A. and K. Louwerse (eds). pp.168–182. IOS Press, Amsterdam, Washington, Tokyo, 1996.
[4] Council of Europe Recommendation R(97)5: *On The Protection of Medical Data.* Council of Europe, Strasbourg, 13 February 1997.
[5] UK Dept. of Trade and Industry ref. URN 97/669: *Licensing of Trusted Third Parties for the Provision of Encryption Services.* London, March 1997.
[6] Menezes, van Oorschot and Vanstone.: *Handbook of Applied Cryptography.* CRC Press, 1996.
[7] Rothermel, K. and R. Popescu-Zeletin (eds).: *Mobile Agents'97 – Proc. 1st International Workshop.* LNCS 1219, Springer-Verlag, April 1997.
[8] Schneier, B.: *Applied Cryptography, Protocols, Algorithms and Source Code in C.* J. Wiley and Sons Inc, 2nd Ed, 1996.
[9] Camp L.J., Sirbu M.: "Critical Issues in Internet Commerce," *IEEE Communications Magazine.* pp.58–62. IEEE Press, 1997.
[10] Yourdon E.: "Java, the Web and Software Development," *IEEE COMPUTER Magazine.* pp.25–30, 1996.
[11] Hamilton M.: "Java and the Shift to Net-Centric Computing,". *IEEE COMPUTER Magazine.* pp.31–39, 1996.
[12] Vigna G. (ed): *Proc. Mobile Agents and Security.* LNCS, Springer-Verlag, 1998 (forthcomming).
[13] Marsh A., Delibasis K., Mouravlianski N. and C. Michael: "EUROMED – A WWW-based multi-mediaTelemedical information system," subm. in *Transactions on Information Technology in Biomedicine.*
[14] Marsh A.: "EUROMED – A WWW-based multi-media medical information system," *Proc. 19th Annual Intl. Conf. IEEE Engineering in Medicine and Biology Society.* IEEE-EMBS, Chicago, 1997.
[15] McGraw G. and Ed Felton: *Java Security: Hostile Applets, Holes, and Antidotes.* J.Wiley, ISBN 0-471-17842-X.
[16] Venners, B.: "Java security: How to install the security manager and customize your security policy,"
<http://www.javaworld.com/javaworld/jw-11-1997/jw-11-hood.html>

[17] Sun Microsystems Inc.: "Secure Computing with Java: Now and the Future," (White Paper) *Java One 1997 Conference.*
<http://java.sun.com/marketing/collateral/security.html>
[18] Sun Microsystems Inc.: "Security-related Java APIs,"
<http://java.sun.com/security/>
[19] Freier, P., Karlton and P. Kocher: "The SSL Protocol Version 3.0," *Internet Engineering Task Force: Internet Draft.*
<ftp://ietf.org/internet-drafts/draft-ietf-tls-ssl-version3-00.txt>
[20] Rescorla, A. and Schiffman: "The Secure HyperText Transfer Protocol," *Internet Engineering Task Force: Internet Draft.*
<ftp://ietf.org/internet-drafts/draft-ietf-wts-shttp-03.txt>
[21] Rivest, R.: "The MD5 Message-Digest Algorithm," MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992. *Internet Engineering Task Force: Request For Comments RFC1321.*
<http://ds.internic.net/rfc/rfc1321.txt>
[22] EUROMED, ISIS '95, DG III programme, 1995-1998.
<http://euromed.iccs.ntua.gr>
[23] EUROMED-ETS: *Trusted Third Party Services for Health Care in Europe.* INFOSEC programme, DG XIII, 1997.
<http://narcisus.esd.ece.ntua.gr/ www/ETS>
[24] NIST, FIPS PUB 180-1: *Secure Hash Standard.* National Institute of Standards and Technology, U.S. Dept. of Commerce, April 1995.
[25] Krawczyk, H., Bellcare M. and R. Canetti: "HMAC: Keyed-Hashing for Message Authentication," *Internet Engineering Task Force: Request for Comments.*
<http://ds.internic.net/rfc/rfc2104.txt>
[26] ANSI X3.106: *American National Standard for Information Systems Data Link Encryption.* American National Standards Institute, 1983.
[27] Tuchman, W.: "Hellman Presents no Shortcut Solutions to DES," *IEEE Spectrum.* 16:8, July 1979.
[28] Thayer, R. and K. Kaukonen: "A Stream Cipher Encryption Algorithm," *Internet Engineering Task Force: Internet Draft,* July 1997.
<http://ietf.org/internet-drafts/draft-kaukonen-cipher-arcfour-01.tx
t>
[29] RSA Laboratories: "PKCS #12: Personal Information Exchange Syntax Standard," (version 1.0 Draft), April 1997.
[30] CEC COM(90) 314 final SYN 287: "On the Protection of Individuals in Relation to the Processing of Personal Data," *Commission of the European Communities,* Brussels, September 1990.
[31] CEC COM(90) 314 final SYN 288: "On the Protection of Personal Data and Privacy in the Context of Public Digital Telecommunication Networks," *Commission of the European Communities,* Brussels, September 1990.
[32] CE R(81)1: *Recommendation R(81)1 on Automated Medical Data Banks,* Council of Europe Convention 108, January 1981, ISBN 92-871-0022-5.
[33] EU 95/46/EC: *On the Protection of Individuals with regards to the Processing of Personal Data and on the Free Movement of Such Data,* European Union Directive, OJ L281/31-50, October 1995.
[34] Simitis, S.: "Reviewing Privacy in an Information Society," *Univ. Pennsylvania Law Review,* V.135, pp.707–746, March 1987.
[35] Sun Microsystems Inc.: "Java Naming and Directory Interface,"
<http://java.sun.com/products/jndi/>

[36] Wahl, M., Howes, T. and S. Kille: "Lightweight Directory Access Protocol (v3)," *Internet Engineering Task Force: Request For Comments RFC2251.* <http://ds.internic.net/rfc/rfc2251.txt>