www.euro6ix.net

Title:	Document Version:
Technical Report TR4.1A.4 **PKIv6**	0.7

Project Number:	Project Acronym:	Project Title:
IST-2001-32161	Euro6IX	European IPv6 Internet Exchanges Backbone

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
31/12/2002	25/02/2003	R – PU

* Type: P – Prototype, R – Report, D – Demonstrator, O – Other
** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	Organization:	Contributing WP:
Antonio F. Gómez-Skarmeta	UMU	WP4

Authors (organizations) in alphabetical order:

Miguel Ángel Morales (Consulintel), César Olvera (Consulintel), Álvaro Vives (Consulintel), Gabriel López (UMU), Gregorio Martínez (UMU), Javier Sedano (UPM).

Abstract:

This deliverable document summarizes all the tasks planned to be done for the first year in the context of the Euro6IX project regarding the design and deployment of the PKIv6 service. During this period a basic PKI has been updated to support IPv6 communications and to manage advanced certification services and protocols. This PKIv6 service is a key element for the deployment of future service and applications with security requirements. Thus, this is considered as an important service to be deployed in ISPs and IXs involved entities.

Keywords:

Attribute Certificates, Certificate, Euro6IX, IPSec, OCSP, PKI, PKIv6, SCEP, SSL, TSP, VPN, X509, 6WIND

Revision History

Revision	Date	Description	Author (Organization)
v0.1	20/05/2002	Document creation	Antonio Skarmeta (UMU) Gabriel López Millán (UMU)
v0.2	19/06/2002	Document updated	Antonio Skarmeta (UMU) Gabriel López Millán (UMU)
v0.3	20/06/2002	Document updated	Antonio Skarmeta (UMU) Gabriel López Millán (UMU)
v0.4	15/07/2002	Document updated for month 6 deliverable	Antonio Skarmeta (UMU) Gabriel López Millán (UMU)
v0.5	27/11/2002	Document updated	Antonio Skarmeta (UMU) Gabriel López Millán (UMU)
v0.6	13/12/2002	Document finished	Antonio Skarmeta (UMU) Gabriel López Millán (UMU) Gregorio Martínez Pérez (UMU)
v0.7	25/02/2003	Logos added an PDF generated	Jordi Palet (Consulintel)

Executive Summary

This draft deliverable summarizes all the tasks planned for the first year of Euro6IX project in the context of WP4 subactivity A4.1-4 (PKIv6). In this period, a PKI solution called PKIv6, has been released supporting a wide group of security services. This solution can now be used by the project partners to establish trust relationships between them and to offer security mechanisms to their end users

Section 1 summarizes what Public Key Infrastructure is and why it should be porting to support IPv6. Section 2 and 3 describe the analysis and design required to achieve this porting and Section 4 shows a first PKIv6 implementation inside the Euro6IX project. Section 5 describes initial designs to deploy PKIv6 in Euro6IX ISPs/IXs context. Section 6 adds a new extension to this, supporting Attribute Certificates and Section 7 shows the advantage of using IPv6.

Table of Contents

Table of Figures

1. INTRODUCTION

1.1 The concept of Public Key Infrastructure

The target of a PKI [PKI] is to provide Public Key Certificate (PKC) management to the group of security protocols designed to protect Internet. These protocols, as IPSec (Internet Protocol Security), SSL (Secure Sockets Layer), TLS (Transport Layer Security) or S/MIME (Secure Multipurpose Internal Mail Extensions) use public key cryptography to provide services such as confidentiality, data integrity, data origin authentication and non-repudiation.

Users of public key based systems must trust in a PKC. It is a data structure which binds a public key to the user subject. This binding is achieved by having a trusted CA that verify the subject identity and digitally sign each digital certificate.

A PKI is defined as a based on public key cryptography system, include software, people, policies, hardware, etc, allowing create, manage, store, distribute and revoke public key certificates. The main components of a generic PKI are:

- Certification Authorities (CAs) issue, renew and revoke PKCs.
- Registration Authorities (RAs) authenticate off-line users and add certificate's properties.
- PKI clients can encrypt and sign digital documents.
- PKI clients validate digital signatures from a known public key of a trusted CA.
- Public repositories make available certificates and certificate revocation lists (CRLs).

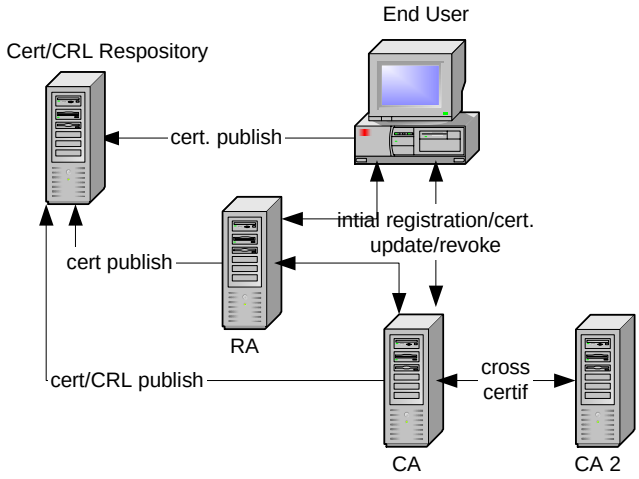

Figure 1-1: PKI Generic Diagram

1.2 Why Porting a PKI to IPv6?

PKI offers services that can be reached via Internet, such as certification requests, retrieval, revocation or renewal PKCs, etc. PKI over IPv6 [IPv6], also called PKIv6, offers the same certification services via IPv6 networks.

Final users will use these services to encrypt, digital sign documents, etc. Besides, services or devices, which need secure communications, can use them. In a IPv6 network there are services, such as secure web servers (HTTPS), Virtual Private Network devices (VPN), Authentication, Authorization and Accounting services (AAA), etc, which can use PKCs to protect information or to user's authentication. Even more, in some scenarios, PKCs can use IP/IPv6 address or a DNS name as an alternative subject, for example, in a HTTPS web server to certificate server authentication. This service is offered by PKIv6, which can include IPv6 addressing inside digital certificates.

Thanks to this infrastructure, users of an IPv6-only network can user cryptographic services to secure their communications.

2. ANALYSIS

2.1 PKIv6 Requirements

PKIv6 developed in Euro6IX project is based on a basic PKI born in the University of Murcia. This source design gave as result a PKI, over IPv4 networks allowing certification, renewal and revocation services for final users and VPN devices.

Inside the Euro6IX project, original PKI has been extended with several services and, of course, with IPv6 support.

2.2 JCA/JCE JDK1.4 Providers

The Java Cryptography Architecture [JCA] was introduced in the first release of JDK Security in JDK 1.1. The JCA is a framework for accessing and developing cryptographic functionality for Java platform. It includes APIs for digital signatures and message digest, also, includes a Provider architecture that allows for multiples and interoperable cryptography implementations.

The Java Cryptography Extension [JCE] is a extension for JCA API which includes APIs for encryption, key exchange and Message Authentication Code (MAC). Both, the JCA and JCE provide a complete, platform-independent cryptography API.

All PKIv6 components are developed in Java language with JDK1.4 version and they should be able to carry out cryptographic operations as encryption, signed, key exchange, etc. For this reason is necessary a JCA/JCE provider which offers this cryptography API.

Nowadays there are many JCA/JCE cryptographic providers. Next, there is a brief summary of some of them.

2.2.1 Analyzed Providers

a) IAIK-JCE – Institute for Applied Information Processing and Communications

The IAIK Java Cryptography Extension (IAIK-JCE) is a set of APIs and implementations of cryptographic functions, including symmetric, asymmetric, stream, and block encryption methods. It supplements the security functionality of the default Java JDK 1.1.x / JDK 1.4, which itself includes digital signatures (DSA) and message digests (MD5, SHA).

Version	3.0
Developer	Technical University of Graz (http://www.tugraz.at).

Features	• Re-implementation of the JCE 1.2 from SUN
	• ASN1 library
	• PKCS standards support:
	  • PKCS#1: RSA Encryption Standard
	  • PKCS#5: Password-Based Encryption Standard

	• PKCS#7: Cryptographic Message Syntax Standard • PKCS#8: Private-Key Information Syntax Standard • PKCS#9: Selected Attribute Types • PKCS#10: Certification Request Syntax Standard • PKCS#12: Personal Information Exchange Syntax Standard • X.509 • X.509 public key certificates • X.509 qualified certificates • X.509 attribute certificates • X.509 and CRL extensions • Netscape Certificate extensions • Extensions for qualified, attribute and OCSP certificates • Implements Online Certificate Status Protocol, v2.01 • Self-designed random number generators • S/MIMEv3 support with IAIK-S/MIME product • SSL/TLS with ISASILK toolkit
JDK1.4	Yes
License	Developers License, Free for educational use
Open Src.	No
URL	http://jcewww.iaik.tu-graz.ac.at

b) JCSI – Java Crypto and Security Implementation

JCSI implements standards-based security services in a manner compatible with the Java Cryptographic Architecture (JCA), enabling easy integration of such services with the Java platform.

Version	2.1.1
Developer	Wedgateil Communications Pty. Ltd.
Features	• JCE 1.2 framework implementation • JCA/JCE crypto provider • ASN1 library • PKCS standards support: • PKCS#7: Certification Request Syntax Standard • PKCS#8: Private-Key Information Syntax Standard • PKCS#10: Certification Request Syntax Standard • PKCS#12: Personal Information Exchange Syntax Standard • X.509 • X.509 public key certificates • X.509 qualified certificates • X.509 attribute certificates

	• X.509 and CRL extensions • Netscape Certificate extensions • Kerberos 5 library • Cryptographic Message Syntax (CMS) • S/MIME v3 library requiring JavaMail 1.2 • SSLv3 and Transport Layer Security (TLS) library implementing JSSE
JDK1.4	No
License	Free for no commercial use
Open Src.	No
URL	http://www.wedgetail.com

c) Bouncy Castle Crypto APIs

The Bouncy Castle Crypto APIs consist of a lightweight cryptography API in Java, a provider for the JCE and JCA, a clean room implementation of the JCE 1.2.1, generators for Version 1 and Version 3 X.509 certificates and PKCS12 files and a signed jar version suitable for JDK 1.4 and the Sun JCE.

Version	1.13
Developer	Legion of the Bouncy Castle

Features	• JCE 1.2 framework implementation
	• JCA/JCE crypto provider
	• ASN1 library
	• PKCS standards support:
	  • PKCS#5: Private-Key Information Syntax Standard
	  • PKCS#7: Private-Key Information Syntax Standard
	  • PKCS#8: Private-Key Information Syntax Standard
	  • PKCS#10: Certification Request Syntax Standard
	  • PKCS#12: Personal Information Exchange Syntax Standard
	• X.509
	  • X.509v3 public key certificates
	  • X.509v3 and CRLv2 extensions
	• Cryptographic Message Syntax (CMS)
	• S/MIME v3 library requiring JavaMail 1.2

JDK1.4	Yes
License	Free for commercial and no commercial use
Open Src.	Yes
URL	http://www.bouncycastle.org

d) BEEJCE/BEECrypt

BeeJCE is Virtual Unlimited's cleanroom implementation of Sun's Java Cryptography Extensions, version 1.2. Because Virtual Unlimited is a Dutch company, developers can use BeeJCE with BeeCrypt for Java to build secure Java products that are freely exportable worldwide. Like BeeCrypt, BeeJCE is available under the GNU LGPL license.

BeeCrypt is an open source cryptography library released under the GNU LGPL license. It contains implementations of many proven algorithms, including Blowfish, MD5, SHA-1, and ElGamal. BeeCrypt is not designed to solve one specific problem, like file encryption, but to be a general purpose toolkit that can be used in a variety of applications. Two versions of BeeCrypt are available: one written in C and assembler and another written in pure Java.

Version	BeeJCE 1.0.2, BeeCrypt 2.2.0
Developer	Virtual Unlimited (VU), http://www.virtualunlimited.com

Features	• Implementation of the JCE 1.2 from SUN
	• ASN1 library
	• PKCS standards support:
	  • PKCS#1: RSA Encryption Standard
	  • PKCS#5: Password-Based Encryption Standard
	• X.509
	  • X.509 public key certificates

JDK1.4	No
License	Free for commercial and no commercial use
Open Src.	Yes
URL	http://www.virtualunlimited.com/products

e) Baltimore Keytools

Baltimore KeyTools enable developers to focus on customer aspects of application functionality, rather than complex security mechanisms and protocols. This allows developers to effectively deliver business-critical applications in the shortest possible timeframe, and without the need to learn complex PKI coding.

Version	5.0
Developer	Baltimore Technologies plc, http://www.baltimore.com
Features	• Cryptographic and digital certificate support • Certificate request and retrieval from a CA • LDAP directory support

	• Central policy control • Smart-Card support • Additional Certificate Authority transporters • OCSP y CRL distribution point support • S/MIME, SSL y XML
JDK1.4	No
License	Developers License, Free for educational use
Open Src.	No
URL	http://www.baltimore.com/keytools/crypto

f) Phaos Crypto Toolkits

Provides the core algorithms for cryptography in Java. Phaos Crypto includes Java implementations of the RSA public key cryptosystem, the ARCFOUR (RC4) stream cipher, DES and triple-DES encryption, MD5 and SHA message digests, Diffie-Hellman Key Agreement, and much more. With Phaos Crypto, you'll write once, and run on any version of Java, from JDK 1.1 to JDK 1.4

Version	5.0
Developer	Phaos Technology, http://www.phaos.com

Features	• JCE 1.2 framework implementation
	• JCA/JCE crypto provider
	• ASN1 library
	• PKCS standards support:
	  • PKCS#1: Certification Request Syntax Standard
	  • PKCS#5: Certification Request Syntax Standard
	  • PKCS#7: Certification Request Syntax Standard
	  • PKCS#8: Private-Key Information Syntax Standard
	  • PKCS#10: Certification Request Syntax Standard
	  • PKCS#12: Personal Information Exchange Syntax Standard
	• X.509
	  • X.509 public key certificates
	  • X.509 and CRL extensions
	• Cryptographic Message Syntax (CMS)
	• S/MIME v3 library requiring JavaMail 1.2
	• SSLv3 and Transport Layer Security (TLS) library implementing JSSE

JDK1.4	Yes
License	Evaluation License
Open Src.	No
URL	http://www.phaos.com/e_security/prod_crypt.html

g) Summary

	Version	OpenSrc	JDK1.4	SSL	SMIME	X509	CRL	OCSP	CMS
IAIK	3.0	no	yes	yes	yes	yes	yes	yes	no
JCSI	2.1.1	no	no	yes	yes	yes	yes	no	yes
BC	1.13	yes	yes	yes	yes	yes	yes	no	yes
Bee	1.02	yes	no	no	no	yes	yes	no	no
Keytools	5.0	no	no	yes	yes	yes	yes	yes	no
Phaos	-	no	yes	yes	yes	yes	yes	no	yes

Figure 2-1: JCA/JCE Summary

Only solutions with JDK1.4 support are valid to use in PKIv6 development. The selected option was IAIK since it is compliant with all requirements and it was used in the original PKI.

2.3 PKIv6 IETF Required Protocols

In this chapter a definition of the main IETF used protocols in PKIv6 is given next:

- SSL (Secure Sockets Layer) [SSL]: This security protocol provides communications privacy over Internet. The protocol allows client/server applications to communicate in a way designed to prevent eavesdropping, tampering, or message forgery.
- LDAP (Lightweight Directory Access Protocol) [LDAP]: Protocol for accessing online directory services. It runs directly over TCP, and can be used to access a standalone LDAP directory service or to access directory service that is back-ended by X.500.
- SCEP (Simple Certificate Enrollment Protocol) [SCEP]: The goal of SCEP is to support secure issuance of certificates to network devices, using existing technology whenever possible. The protocol supports operations of CA and RA public key distribution, Certificate enrollment, Certificate revocation, Certificate query and CRL query.
- OCSP (Online Certificate Status Protocol) [OCSP]: This protocol enables applications to determine the status of a digital certificate, that is to say, if certificate is revoked or not. OCSP may be used to satisfy some of operational requirements of providing more timely revocation information than is possible with CRLs and may also be used to obtain additional status information.
- TSP (Time Stamp Protocol) [TSP]: This protocol allows to prove that a datum existed before a particular time and can be used as a Trusted Third Party as one component in building reliable non-repudiation services.
- S/MIME (Secure Multipurpose Internal Mail Extensions) [SMIME]: Provides a consistent way to send and receive secure MIME data. Based on the popular Internet MIME standard, S/MIME provides the following cryptographic security services for electronic messaging applications: authentication, message integrity and non-repudiation of origin (using digital signatures) and privacy and data security (using encryption).
- IPSec (Internet Protocol Security) [IPSEC]: Used to provide privacy and authentication services at the IP layer.

3. DESIGN OF PKIV6

The Public Key Infrastructure with IPv6 support designed is based on the design and later implementation of a complete and robust group of certification services for any organization type that wants to provide its clients and/or users of security mechanisms in their communications.

By means of their services, an user will be able to carry out any operation type from his own navigator: to request a certificate, renewal or revoke it, to look for another user certificate which wants to establish a secure communication, etc. Moreover, it allows users use of smart cards (which can be distributed by the own organization) to store cryptographic information, so that it facilitates the mobility of these.

PKIv6 is an infrastructure designed completely in Java technology, what allows its use in any platform. Until the moment, use of smart cards is limited to Windows environments, but our intention is to apply this characteristic to Unix systems.

The main characteristics are:

- Users can issue, renew and revoke certificates.
- LDAPv6 directory supported to store users and CA's certificates and CRLs.
- Final users can carry out certification operations from their own navigator or through RAs.
- Users can storage cryptographic information (private key, certificate and CA's certificate) in their smart cards. This allows total mobility, so that, if an user requests a certificate from a navigator or from the RA, this certificate can be recovered in any moment from another different navigator.
- Basic configuration of PKIv6 through HTTP.
- Policy definition will establish the opportune restrictions inside an organization. Administrators will define this Policy and it will be applicable in all the PKI's c omponents.
- PKIv6 has been developed completely in Java, what allows use of any platform in the system. It is based on standards specified by the IETF inside the PKIX [PKI] work group.
- SCEP protocol supported (Simple Certificate Enrollment Protocol) for VPN Clients.
- 6WIND VPN routers supported.
- Cross-certification is allowed in two ways, peer-to-peer and hierarchical cross-certification
- Communications between components are over IPv6 or IPv4.

3.1 Components

The following diagram shows PKIv6 main components:

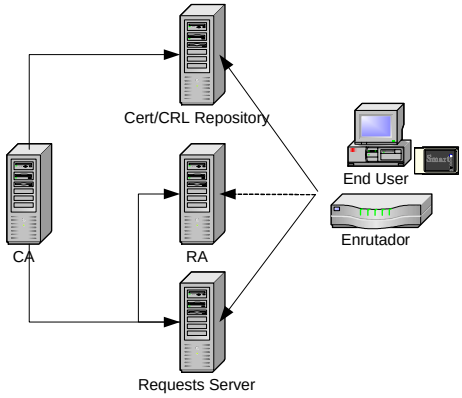

Figure 3-1: PKIv6 Main Components

3.1.1 Registration Authority

RA is the component that takes charge of sending requests to be processed by a CA. The system can be formed by several RAs. The Administrator will be able to carry out certification, renewal and revocation requests and he will be able to check the local Policy. RA, in Windows environments (at the moment), supports the full use of smart cards.

3.1.2 Requests Server

RQ is the component that picks up all requests from the system's entities. It picks up the RA, final users and administrators requests. These are stored in an internal database to be processed by the CA.

3.1.3 Certification Authority

CA processes all requests stored in the RQ Server. If it is a certification request, and the certificate can be issued, this is stored in an internal database, made public in the LDAP repository and the user requested them is notified through a signed email. If it is a renewal request the certificate is updated in the internal database and in the LDAP repository, and if it is a revocation request the certificate is marked as revoked in the internal database and it will be included in next published CRL.

It is important to highlight RQ Server and CA connection is unidirectional, so that, CA does not have incoming connections in any moment.

3.1.4 Final Users

Besides through RA end users can achieve certification operations through a web browser. If it is a certification request this is stored in the RQ Server and it will be validated or eliminated by the RA. If it is a renewal or revocation request this will be treated directly by the CA. Users that want to operate through web browser can make use of their smart card to store their cryptographic information thanks to a CSP. This facilitates users mobility; for example, a user can request a new certificate from his web browser or a RA and the private key is stored in his smart card, later on he can recover the associated certificate and CA's certificate from any other navigator.

3.1.5 Policy

Policy definition is one of the main characteristics in PKIv6. Operations achieved by PKIv6 components should be compatible with the Policy and they will not be processed otherwise.

Policy defines the main restrictions an organization could apply in the operations of their users. For example an organization could require all issued certificates use RSA keys, with 2048 bits key length, or to force a certificate to be renewal inside some certain terms of time.

3.1.6 Certificates Repository

PKIv6 supports, in an optional way, the use of certificates repository. In a LDAPv6 directory all user's certificates, CA's certificate and CRLs can be stored so that they can be consulted by users.

3.1.7 Smart Cards

PKIv6 supports use of smart cards, in RA and CA entities and, also, in web environments. Users will be able to have smart cards to store their cryptographic information. Each smart card will contain the user's private key, associate certificate and CA's certificate.

3.2 Basic Services

Basic services are in most PKIs and manage certificate's cycle of life, that is to say, certification requests, validation, publication, revocation and renewal requests. Next, these are explained.

3.2.1 Certification Request

Certification Requests can be done of several ways:

- In the first way, users can make their request going to a RA with personal documents to prove their identity and, if it is used, their smart cards. RA issue a new request composed by a PKCS#10 object, it is send to RQ in a secure way through SSL connection and the associated private key is stored in a file or in the user's smart card.
- In the second way, services administrators can go to a RA with a previously generated PKCS#10 requested with another software. This request should be authenticated by the administrator and by the system's Policy and, if validation is correct, request will be send to RQ to be processed. Besides, applications/services can be certified through SCEP protocol, which allows devices to carry out certification operations. VPN devices usually use this service.
- In the last way, users can issue a new certification request through web using their own browser and, if it is used, their smart card. In this way, users send requests to RQ, RA's administrator recovers these requests and validate them. Then the requests are sent to RQ again to be processed by the CA.

3.2.2 Retrieve a Digital Certificate

Once CA has issued a certificate, user should retrieve it. In this infrastructure, retrieval is done through a web browser or LDAP repository. The user's certificate, CRL and CA's certificate can be imported into the user's browser through web, besides, if user has a smart card, it can be used to store private and public key thanks to the PISCIS Cryptographic Service Provider [CSP].

Cryptographic devices, like VPN devices, can retrieval certificates directly from RQ. Besides, valid certificates are stored in a public repository and can be retrieval by any LDAP client.

3.2.3 Renewal Request

Final entities can renew their certificate's validity period according to the system's policy. This request can be done going to RA or by means of the web site through SSL connection.

3.2.4 Revocation Request

In exceptional conditions, as lost or compromised of smart card, digital certificates are invalidated before validity period expires. Users have two options to revoke their certificates: to go to RA and request new revocation or to issue this in the web site. The last option is only possible whether user even has his certificate's private key.

3.2.5 Policy Definition

Policy is a digital document containing a serial number, issue and next issue date and a group of policy elements. This group of elements point out that policy's restrictions should be applied to that group of certificates. For example, Policy elements will set RSA key size for all identities belonging to group "ou=MyOrganizationalUnit, o=MyOrganization, c=MyCountry".

Once established, Policy is signed by CA and will be accessible for all PKI components.

3.3 Advanced Services

PKIv6 offers several added value services dedicated to enrich the range of possibilities it offers. Among them we find support for SCEP (Simple Certificate Enrollment Protocol) protocol and certification for 6WIND routers.

3.3.1 SCEP

This service gives certification support to VPN devices requiring use of digital certificates. It is a protocol developed by CISCO broadly used by devices as routers to obtain their secure information when carrying out virtual private networks.

Implementation of this service is based on a SCEP server, implemented with Java Servlets which waits certification requests or consults from VPN devices. Current implementation is based on use of single keys and it has been tested successfully in CISCO routers.

3.3.2 6WIND VPN routers

6WIND routers, advanced IPv6 routers, require an special way to obtain cryptographic information. Issuance and retrieval of certificates is based on SSH [SSH] protocol.

PKIv6 has been updated to allow clients making SCP [SSH] operations to require or retrieval certificates. This implementation is based on a Java Servlets making pooling over a special user directory which only 6WIND authorized clients can put and get information. When authorized clients put a certification request, this is validated or not by the RA administrator. When the certificate is issued client can retrieval it from the same directory. Besides, authorized clients can get CA and CRL certificates.

3.3.3 Cross-certification

Cross-certification is a process carried out by Certification Authorities to establish trust relationship between them. When two Certification Authorities are cross-certified all their PKCs and keys trust the other ones like if all belongs to the same organization. To make it possible, two CAs should exchange their cross-certificates. In other words, cross-certification is used to allow client systems or end entities in one administrative domain to communicate securely with client systems or end users in another administrative domain.

There are two types of cross-certification operations. The first one is "peer-to-peer" cross-certification, is the establishment of a trust relationship between two root CAs through signing of another CA's public key in a new certificate, as well-know as a cross-certificate. The second one is the most common way to establish trust between CAs, this way, a root CA establish trust relationship with a subordinate CA and so on. To verify end user's certificate, chain with all CA certificates involved should be verified, this is called "hierarchical" cross-certification.

The following figures show cross-certification options:

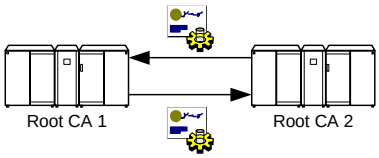

Figure 3-2. Peer-to-peer cross-certification

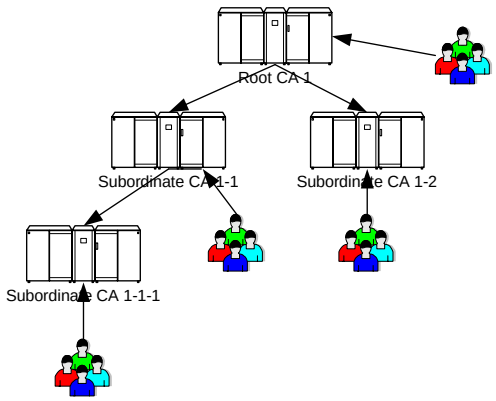

Figure 3-3. Hierarchical cross-certification

PKIv6 has been updated to establish peer-to-peer and hierarchical trust relationship between CAs.

Peer-to-peer cross-certification: PKIv6 allows CAs to sign another CA's certificate to make peer-to-peer cross-certificates. In this way, only root CAs can establish this trust relationship. This communication is bidirectional and one peer-to-peer cross-certificate is generated in each way.

Hierarchical cross-certification: PKI administrator can use CA application to issue new PKCs for subordinate CAs. When a subordinate CA is installed it requires the previous cross-certificate. When end user requires a PKC, certificate chain with all intermediate CAs and root CA is returned. This means all applications verifying end user's certificates should verify all certificates in the chain.

4. A PKIV6 IMPLEMENTATION

4.1 Software Components

4.1.1 Basic Software

All developed software is written in Java. Each component, RQ, RA, CA, etc, is prepared as a independent box. We can found tarballs *euro6ix-pki-v6.XX-YY.tar.gz*, where: XX is software version, and YY is a application identifier (RA, CA, RQ or DB)

4.1.2 Required Software

Required software for all components is Java Development Kit 1.4. Besides, a Requests Server needs to have installed:

- Apache_1.3.19 with IPv6/SSL support
- ApacheJServ_1.1.2
- PostgreSQL 7.2 or higher

Certificate Repository should be LDAP compliant software, such OpenLDAP or iPlanet Directory Server.

4.2 Test-bed Set-up

4.2.1 Main Euro6IX PKIv6

PKIv6 test-bed set-up in UMU is a complete system, which can offers certification services to final users and VPN devices, such an IPSec router. It is made up of a Requests Server for external requests, a Registration and Certification Authorities, and a LDAP repository to retrieve PKCs and CRLs. The following figure shows the system:

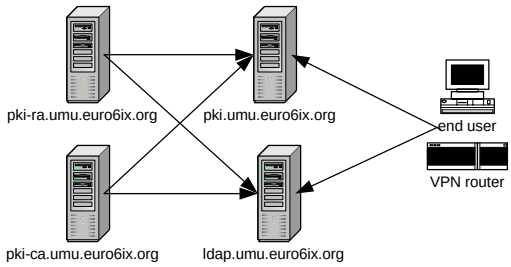

Figure 4-1: A PKIv6 Test-bed

This test-bed uses Linux Red Hat platform in all components. It is not mandatory, and Windows systems could also be installed.

Final users can reach PKIv6 through web browser and VPN devices should use SCEP protocol or 6WIND SCP method to get PKCs. Moreover, users can use smart cards to store their public and private keys. This PKI is also accessible through IPv6 6Bone network.

4.2.2 Cross-certification test-beds

Besides main test-bed, we have deploy two additional test-beds to trial cross-certification implementations.

In first test-bed, a hierarchy of CAs permits to a big organization to delegate management of PKCs into subordinates CAs. The following figure shows a diagram of this infrastructure:

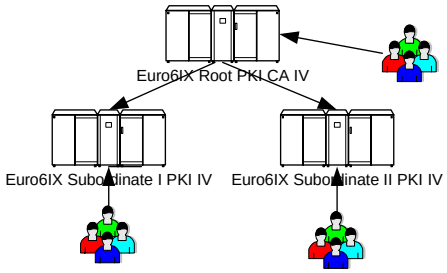

Figure 4-2. Euro6IX hierarchical PKI v6

To allow this configuration, Euro6IX Root PKI CA IV should define two new subordinate CA certificates, like the next figure shows:

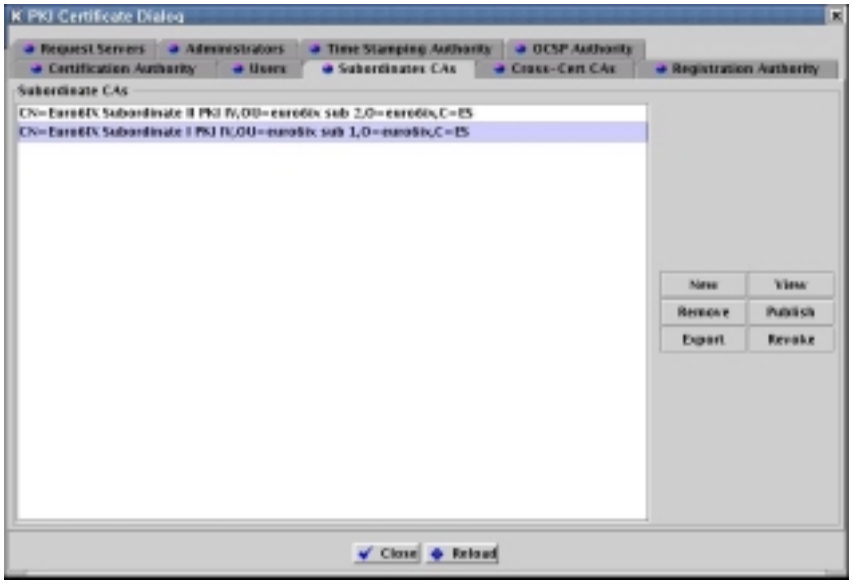

Figure 4-3. PKIv6 Subordinate CAs definition

In second test-bed, a peer-to-peer cross-certification is established between *Euro6IX Root PKI IV* PKIv6 and a new PKIv6, *Euro6IX Root PKI V*. To establish this trust relationship first CA should define a new cross-certificate to the second CA and vice versa.

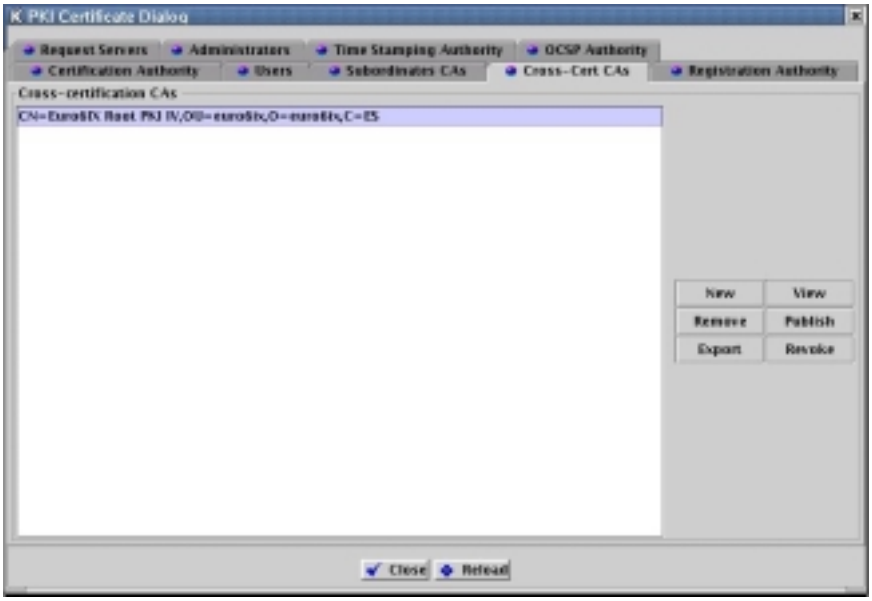

Figure 4-4. Euro6IX peer-to-peer CAs

4.3 Betatesting at UPM

UPM has agreed to betatest the PKI implemented and deployed by the University of Murcia, in order to let it be used by other applications in the project.

The usage of the PKI web page is, from the point of view of the client, pretty easy: after a few clicks, the certificate is requested to the PKI administrator, and when accepted, it can be downloaded directly to the web browser if wanted, or stored to be used in another application.

The certificates provided by the UMU PKI are being used by the clients of the AGWSv6 application (to allow users to log in without needing the user a login/password pair), and by the server itself to authenticate against the clients.

Also, it is being used by Netscape Messenger to provide authentication and ciphering to the mail management

In the future, the PKI infrastructure will be used to deploy IPv6sec tunnels using the CA mechanism, as well as providing security to new applications.

4.4 Pointer to the Euro6IX main PKIv6 Server

Euro6IX main PKIv6 can be reached at the following URL:

https://pki.umu.euro6ix.org

There you can carry out certification requests. If you want to notify a new PKC or do you have any question, please send an email to pkiadmin@dif.um.es.

4.5 User Manual

This User Manual describes different services users can use to make certification operations. These services are offered by a Requests Server allowing final users to do basic services such issue, revoke, renewal or retrieval a certificate, and by a SCEP server, which allows the same operations in a VPN device.

4.5.1 Basic Services

Requests Server

From PKIv6 web site users can achieve a wide group of certification operations. In the URL *https://pki.umu.euro6ix.org* we found the main page with a welcome message, main menu is in the left column.

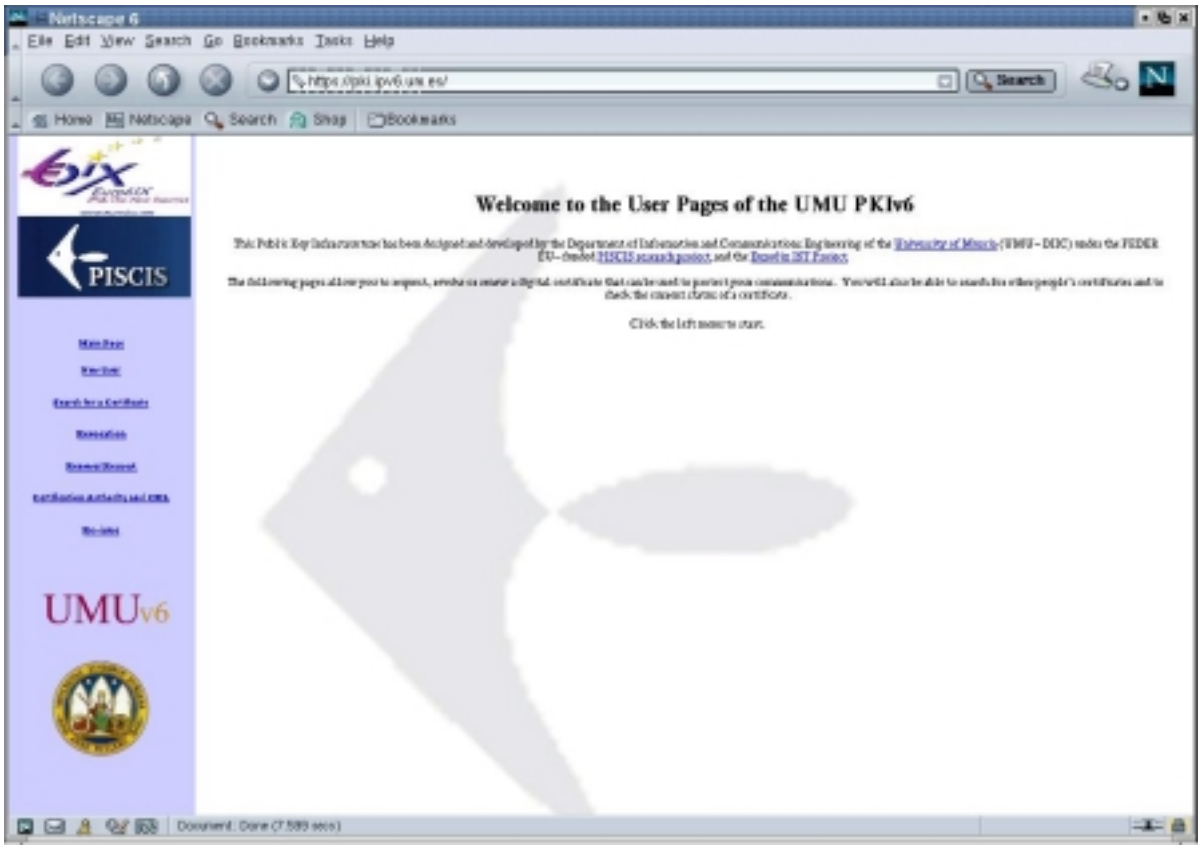

Figure 4-5. RQ Main Page

Main menu offers the following links:

- New User: To request a new certificate.
- Search Certificate: To allow users to search PKCs, the search one can be done by the certificate's Unique Identifier (UID) or serial number.
- Revocation: Show a new page where user can revoke the selected certificate indicating a revocation's reason.
- Renewal: To allow sending renewal requests.
- Certification Authority and CRL: Users can retrieve the CA's certificate and the current CRL. Both can be saved into a file or inside client browser.

When you access to this site a SSL connection requests client certificate, if user doesn't have none yet, this request can be ignored, otherwise he can select anyone. This certificate will be used for revoke and renewal operations.

Next, these operations are described in detail.

New User

Certification request can be done through the web site and they can be divided in two groups: Simple and Advanced.

Simple certification request can be done by Netscape Communitcator 4.7 and higher, or by Microsoft Internet Explorer. In this, user only should indicate the full name, a unique identifier (if required) and the certificate email address. The system's Policy will complete fields as Country, Organization or Unit Organization. If any field is unknown you should send an enquiry to the system administrator. Besides, user should indicate a contact email address or phone for notifications.

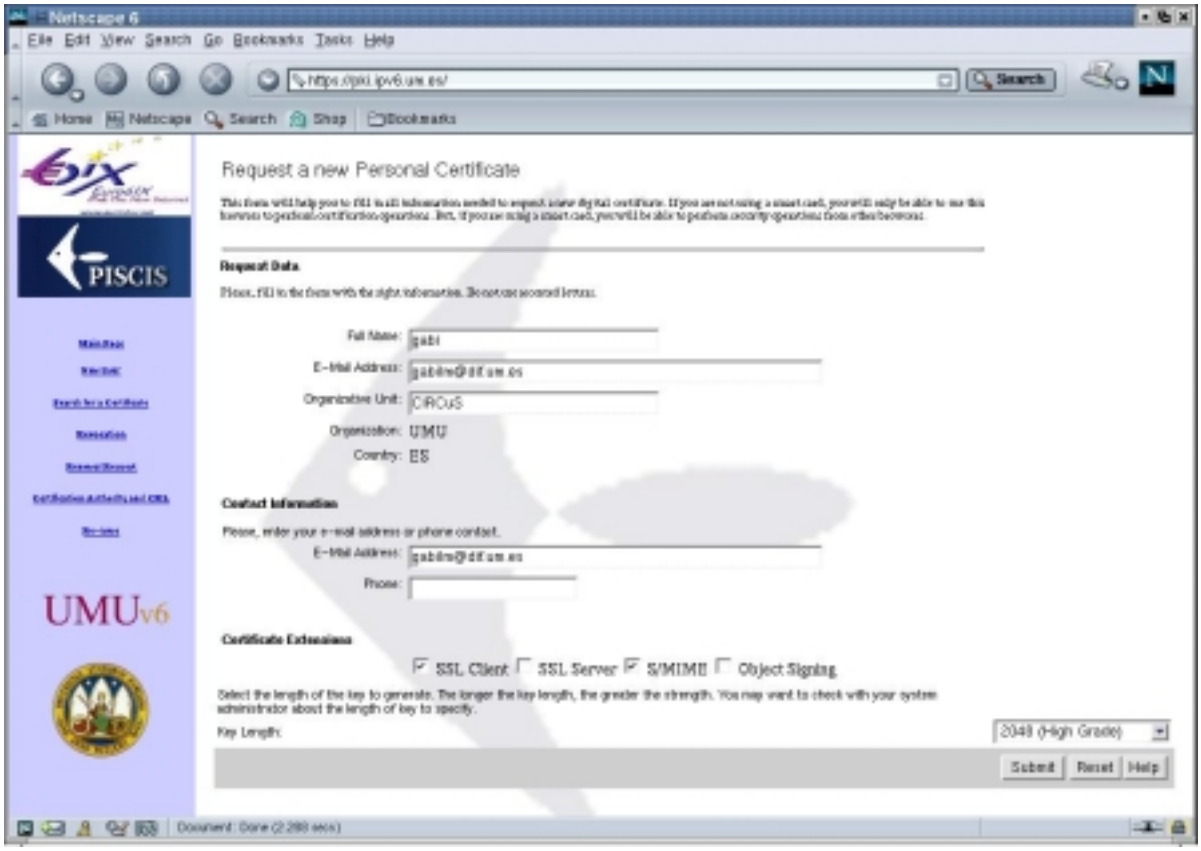

Figure 4-6. RQ New User

When fields are completed user's request can be send to the server. If it is correct, server will show the request number and the user should notify the RA Administrator, which should validate the new request and send it to the CA. When this certificate is issued the user is notified through a signed email or phone contact and his PKC can be retrieve with a web browser.

Advanced certification request allows user to specified the certificate content in detail, as well as the use of smart cards to store public and private keys. Advanced certification is only available on MI Explorer since for cryptographic operations it has been used the PISCIS Cryptographic Service Provider (CSP) [CSP].

Advanced certification requires the same information as simple mode. Besides it allows:

- Add the certificate purpose:
    - SSL client (default).
    - Secure Email (default).
    - SSL server.
    - Object Signing.
- Add a certificate's alternative subject name, for example, an IPv4/IPv6 address or a DNS name.
- To change selected CSP to make cryptographic operations. Besides, for each provider you can choose a key size and a hash algorithm.
- If user wants to use PISCIS CSP, he can define if the private key will be or not exportable and whether the old private key stored in smart card will be removed.

When a public key certificate is requested, user should wait until the notification arrives containing the certificate serial number and how to retrieve it.

Search Certificate

When users want to search their own certificate or another person certificate they should link in *Search Certificate* option. From this page, they can achieve searchs by means of the certificate's serial number, email address or unique identifier (if required). When fields are completed server show a page with the certificate information.

The showed information is:

- Basic certificate fields: version, serial number, issuer, subject, validity period, status, etc.
- Fingerprint.
- Cryptographic Service Provider.
- PEM format certificate which can be exported to a file.

With this information there're two options:

- In one hand, to import the certificate as own. The user should use the browser with which generated the public key or the smart card with the private key associated in a browser with the PISCIS CSP installed.
    - In Microsoft Internet Explorer user can see the certificate in:
        - Tools→ Internet Options → Content → Certificates → Personal
    - In Netscape:
        - Communicator → Tools → Security → Certificates → Your Certificates
- On the other hand, the user wants to import the certificate of another person, he can see the certificate with Microsoft Internet Explorer in:
        - Tools→ Internet Options → Content → Certificates → Other People
    - In Netscape:
        - Communicator → Tools → Security → Certificates → Other People

Revocation Request

To be able to make a revocation request, the user should select, at the beginning, the certificate he wants to revoke. It is necessary because when the user sends the request, the server recovers automatically the user certificate from the web browser. Once the certificate is selected, user should point out the revocation's reason.

The possible options are:

- unspecified
- keyCompromise
- cACompromise
- affiliationChanged
- superseded
- cessationOfOperation
- certificateHold
- removeFromCRL

When the request has been sent, the server responds with a serial number or an error message if there was any problem. When the certificate is revoked the user is notified either by a signed email or phone contact. To check if the certificate has been really revoked user has several ways:

- Search in the server database with the option *Search Certificate*. The certificate will be show as revoked.
- Download laster CRL, It should contain the revoked certificates.

Renewal Request

To be able to make a renewal request user should have selected, at the beginning, the certificate he wants to revoke. It is necessary because when the user sends the request, the server recovers automatically the user certificate from the web browser.

The request will be sent to the server and will be processed by the CA. Once the user is notified he will be able to recover the renewal certificate. The new validity period is automatically set by the system's policy.

Certification Authority´s Certificate and CRL

The CA´s certificate and the Certificate Revocation List should be retrieval from the web site at the beginning. Both can be store in a file, with the option *Save to File*, or can be imported automatically to the system.

To see the certificate with Netscape:

Communicator → Tools → Security → Certificates → Signers

With Microsoft Internet Explorer:

Tools→ Internet Options → Content → Certificates → Authorities

A CRL is a time stamped list identifying revoked certificates that is signed by a CA and made freely available in a public repository such a LDAP directory. In Microsoft Internet Explorer, CRL can not be imported in the system, and in Netscape is imported into the browser and it will be automatically checked every time the browser uses a certificate issuer by the same CA.

4.5.2 Advanced Services

SCEP

SCEP is a CISCO designed protocol to provide secure issuance of PKCs to network devices using standard protocols as PKCS#7 and PKCS#10 [PKCS]. The protocol supports operations of CA and RA public key distribution, certificate enrollment, revocation and query, and CRL query. Such devices as routers, servers, etc. can use SCEP to obtain public and private keys to establish Virtual Private Networks (VPN) in a secure way.

The main steps to access a PKI from a VPN device are:

- Generate Keys
  The public and private keys will be used to create a new PKCS#10 request.
  cisco(config)#crypto key generate rsa
- Configure CA identity
  To configure a new CA, the administrator should associate a new name, if there is an intermediate RA, the access URL to the PKI, LDAP access and if he wants to use or not CRLs.
  cisco(config)#crypto ca identity PKIv6-name
  cisco(ca-identity)#enrolment mode ra
  cisco(ca-identity)#enrolment url http://rq/servlet/piscis.pki.ca.servlets.SCEP
  cisco(ca-identity)#query url ldap://serv_ldap
  cisco(ca-identity)#crl optional
  cisco(ca-identity)#exit
- To obtain CA and RA certificates
  cisco(config)#crypto ca authenticate PKIv6-name

This command shows the CA's certificate Fingerprint.

- To make a certification request
  cisco(config)#crypto ca enroll PKIv6-name
  cisco(config)#exit
  cisco#show crypto ca certificate

The first command sends a new request to the server, after the RA administrator validate this request the new certificate is stored automatically inside the device. The last command shows the system certificates.

6WIND VPN routers

To configure new certificates in a 6WIND router 6 steps are needed:

1- Define a new CA

*nurn{}**ca** PKIv6-name https://pki.umu.euro6ix.org*

2- Install the CA certificate

*nurn{}**import ca_cert** PKIv6-name [cacertremotename]*

3- To get a new certificate

*nurn{}**cert_req** idname PKIv6-name*

This command generates a new request stored in the file idname.req that should be exported to the PKI using the following command:

*nurn{}**export cert_req** idname PKIv6-name*

4- The PKIv6 should issue the new certificate

5- The certificate should be imported into the router

*nurn{}**import cert** idname PKIv6-name [certremotename*]

6- To make the certificate usable in VPN you should select an identity and a trusted CA as follow:

*nurn{myconfig-sec}**ike_id** idname*

*nurn{myconfig-sec}**trust** PKIv6-name*

4.6 Test-bed Final Users and Client Systems

The objective of this infrastructure is to provide public key certification services to the Euro6IX project members. It does include to final users and the project activities that require these services such activities A4.1 and A4.2.

Potential clients of PKIv6 should be end users, VPN devices for IPv6 network security, Mobile and Multicast nodes in advances IPv6 networks, AAAv6 services and Multimedia applications as ISABELv6 or AGWS. The administrators of these applications can request new public key certificates in the main PKIv6 Home Page URL or they can inquire any information on the PKIv6 administrator email address.

Information about how PKIv6 could be installed in the Euro6IX project environment is described in chapter 5.

5. INITIAL DESIGN TO DEPLOY A PKIV6 INFRASTRUCTURE IN THE EURO6IX ISPS/IXS ENTITIES

Design of a PKIs infrastructure inside the Euro6IX project is a hard task requiring a deep study of each partner and IX's requirement and hardships. This chapter is a description of the several PKIv6-related solutions that we envisage, although further discussions are needed.

5.1 Case 1. An Euro6IX Root CA

This design establishes a main root CA for all entities in the Euro6IX project that want to define a Subordinate CA and to generate end user public key certificates. This is a two levels hierarchical cross-certification. All those entities (ISPs or IXs) wanting to be a Subordinate CA could be it.

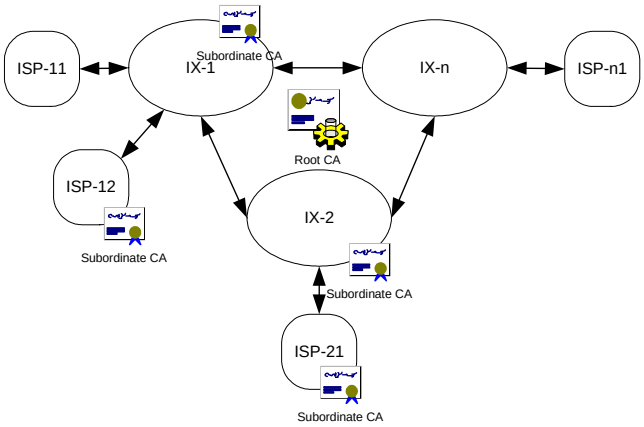

Figure 5-1. Main root CA

- Advantages:
  - Centralized model, there is only a root CA to all participants.
  - Automatic trust relationship between partners.
- Disadvantages
  - All participants depend of one point, the root CA.
  - There is not distinction between IX and ISP
  - Only one root CA for all the project.

The next figure shows an example of deployment inside the Euro6IX project, supposing:

- IX: MAD6IX, and UK6IX
- ISP: UMU, TID, BT or Consulintel.

In this example, root CA is physically placed in an ISP entity.

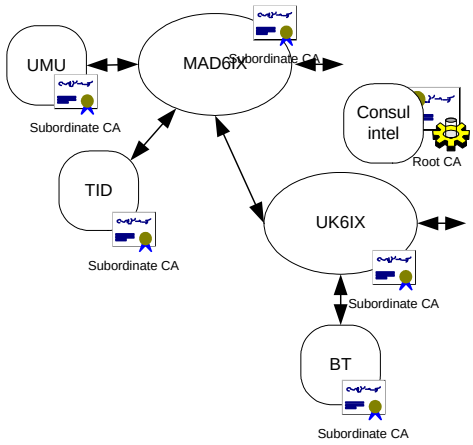

Figure 5-2. Main Euro6IX root CA

It could be a good solution in a research environment, but it seems not to be useful in a real ISP/IX environment.

5.2 Case 2. Root CAs in IX

In this case, only IXs have a root CA and their associated ISPs should have a Subordinate CA. This solution is based on peer-to-peer cross-certification between IXs and one level's hierarchical cross-certification between IXs and their associated ISPs.

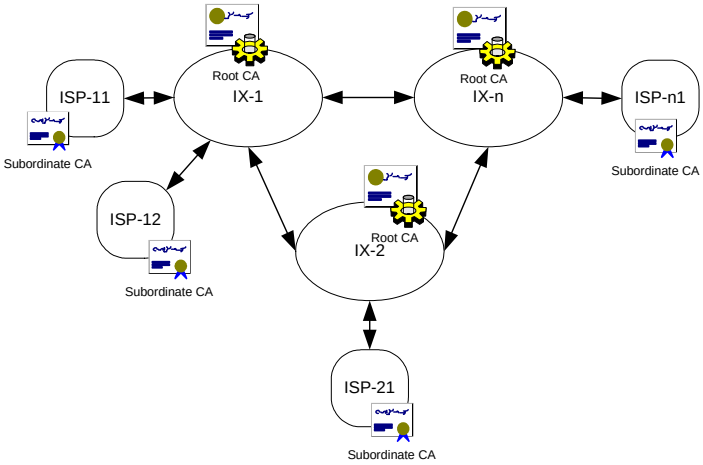

Figure 5-3. Root CAs in IXs

- Advantages:
  - Administrative domains can be defined based on IXs.
  - More flexible, there is not a centralized point.
  - Better scalability, defining domains based on IXs.
  - IXs get to be independent.
- Disadvantages:
  - Trust relationship should be defined between IXs to obtain a full trusted environment.
  - ISPs depend on their associated IX to deploy a subordinate CA.

Following the last example, the next figure shows this case inside the Euro6IX project.

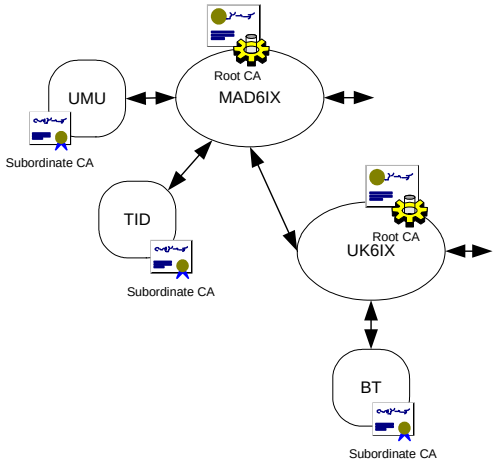

Figure 5-4. Root CAs in Euro6IX IXs

This solution is more inflexible and scalable; besides, there is a close relationship between ISPs and IX. It could be a good solution in less restrictive environments.

5.3 Case 3. Only Root CAs

The Case 3 proposes each entity could be a root CA. Trust relationship can be established between two CAs using peer-to-peer cross-certification. There is not hierarchical cross-certification.

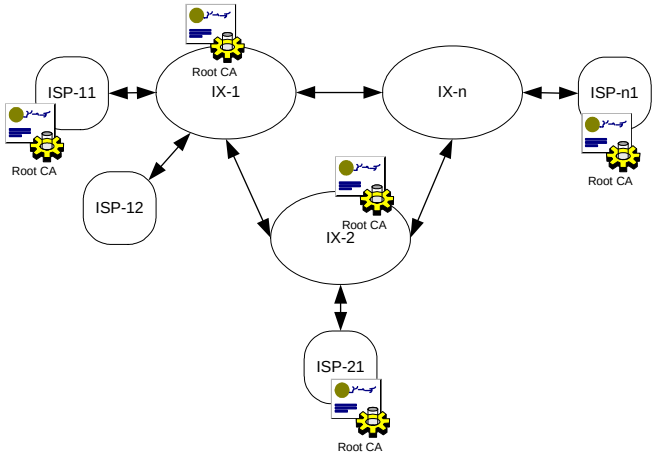

Figure 5-5. Only root CAs

- Advantages:
  - Entities are completely independent.
  - The most flexible. Each entity can be a root CA.
- Disadvantages:
  - The less scalable one. To obtain a full trusted environment each entity should establish a trust relationship with all the others entities.
  - It is necessary a high number of peer-to-peer cross-certification.
  - Relationship between ISPs and IX gets lost.

The next figure shows this case inside the Euro6IX project:

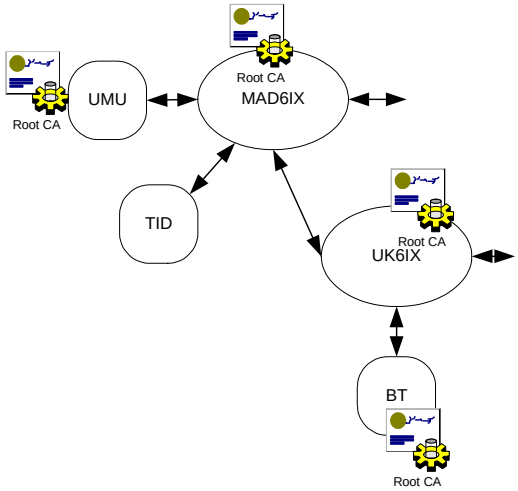

Figure 5-6. Only root Euro6IX CAs

More difficult to manage in a research environment but more realistic in enterprise environments.

6. ATTRIBUTE CERTIFICATES

6.1 Description

Nowadays, a PKI only manages identity certificates, which hold a public key linked with the user's identity. With this kind of certificates an user can be identified, but sometimes it is not enough and we should determine what kind of operations users can achieve. Identity Certificates used by the actual PKIs can not provide these services. That is why the IETF Work Group has added a new kind of certificates, Attribute Certificates [PKI].

An Attribute Certificate (AC) is a signed sentence adding additional properties to an Identity Certificate. This allows extending these certificates to include a group of attributes that can supply privileges. Therefore, an Attribute Certificate is linked with one or several Identity Certificates that identify its owner. Examples of systems using Attribute Certificates could be an access control system based on public key cryptography or Pay Per View services.

We have proposed a basic implementation of an Attribute Certificates system with the following characteristics:

- Certification and renewal options.
- LDAPv6 directory support to store certificates.
- Smart Cards support
- Policy definition establishing privileges granting.
- It is developed in Java language.
- Communication between components is over IPv6 or IPv4.

6.2 Components

We have deployed an extension to PKIv6 which manage Attribute Certificates to end users as well as define user's privileges. The system is based on an Attribute Authority entity (AA) which issues the end user's Attribute Certificates and an Access Point entity (AP) where end users request these certificates. The following figure shows this infrastructure:

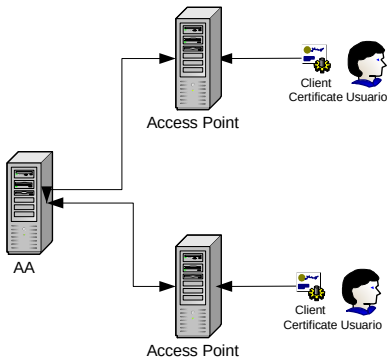

Figure 6-1. Attribute Certificates infrastructure

In this design, to request a new AC end users should present a valid Identity Certificate previously generated with a PKI, like PKIv6. If it is a valid certificate, new request will be issued and if it is Policy compliant the new AC will be linked to this one holding the new privileges. The following figure shows the form to request a new AC:

Figure 6-2. AC Request

When a new AC is requested attributes can be added in the following form:

Figure 6-3. AC Attributes

6.3 ACs in PKIv6

This solution of Attribute Certificates is a PKIv6's added value service which allows extending the properties and the functional character of PKIs. This AC infrastructure could be used joined with PKIv6 to allow end users, like ISPs final users or IXs administrators, to grant access to restricted services, pay peer view services, control access to secure sites, personalized information, etc.

7. IPV6 & IPV4 CONSIDERATIONS

As explained in chapter 1.2 a PKI over IPv6 communications is necessary to offer security services to final users/entities using IPv6-only communications.

One of the most important advantages is the access to digital certificates to entities wanting to create VPN using IPSec/IPv6. In IPv4, IPSec can be used patching the IP stack to support IPSec options, but in IPv6 IPsec is mandatory to be supported. Besides, now it will allow communications between PKIv6 components to be at network level using IPSec and not using application level protocols like SSL or SSH. Another important advantage is that entities do not need dual stack IPv4/IPv6 and IPv6-only systems can carry out cryptographic operations with PKIv6.

8. CONCLUSIONS

The Department of Information and Communications Engineering of the University of Murcia (UMU-DIIC) has released the first IPv6-only fully supported implementation of a Public Key Infrastructure. This security service, called PKIv6, is one of the first important results of the Euro6IX IST research project. It has been written in Java using as many open-source components as possible (Apache Web Server, OpenLDAP, Apache JServ, PostgreSQL, and so on). It has some basic components as a certification authority, and one – or several – registration authorities and directory servers. Some other additional PKI-related services have been implemented, such as, time stamp and OCSP servers, SCEP (for router certification) or cross-certification.

This PKI is being currently tested as a basic security service for static and dynamic IPv6 VPNs, Mobile IPv6 implementations and AAAv6 frameworks. It is also being used with high-level applications such as collaborative environments and videoconference applications.

PKIv6 can be used inside the Euro6IX project to establish a full hierarchical CAs infrastructure and to offer security services to all partners involves. It is also an ideal service for ISPs or IXs that want to offer public key certificates and certification services to end users, and to establish trust relationship between them.

9. FUTURE WORK

PKIv6 is a complete and robust system which offers basic and advances certification services, but not everything is still developed. A PKI can be extended in many ways, adding new services, upgrade standards, adding new characteristics, etc. Next paragraphs depict which can be the future work roads for this system.

The first action should be the establishment of real scenarios inside de Euro6IX project, where involved entities can establish trust relationship between them and they can offer certification services to end user, as described in chapter 5.

Other basic services should be included, like DNSSec support. In this service a PKI should be able to publish certificates in a DNS environment and users could be able to retrieve them. It is especially useful in ISP environments. It also allows secure communications between DNSSec servers.

Other advanced characteristic should be included, but these require a bigger effort and they are not clear yet. These advanced characteristic are: Double Key support, to allow users obtain different sign and cipher keys, and Private Key storage, to allow administrator store all user private ciphering keys.

10. REFERENCES

[PKI] R. Hously, W. Ford, W. Polk, D. Solo, *"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile"*. Request For Comment (RFC) 3280, April 2002.

[IPv6] IETF IP Version 6 Working Group, [www] http://www.ietf.org/html.charters/ipv6-charter.html, November 2002.

[JCA] Sun Home Page, [www] http://java.sun.com, November 2002.

[JCE] Sun Home Page, [www] http://java.sun.com/products/jce, November 2002.

[LDAP] Boeyen S., Howes T., Richard P., *"LDAPv2 Scheme"*, Request For Comment (RFC) 2587. June 1999.

[SCEP] Liu X. et al., *"Simple Certificate Enrollment Protocol"*, IETF draft. draft-nourse-scep-04.txt, February 2001.

[OCSP] Myers M. et al., *"Online Certificate Status Protocol".* Request For Comments (RFC) 2560 June 1999.

[TSP] Adams C., Cain P., Pinkas D., Zuccherato R., *"Time Stamp Protocol"*. IETF draft. draft-ietf-pkix-time-stamp-15.txt, May 2001.

[SMIME] Dusse R., Hoffman P., Ramsdell B. y Weinsteing J., *"S/MIME Version 2 Message Certificate Handling"*, Request For Comment (RFC) 2312, March 1998.

[IPSEC] IETF IP Security Protocol, [www] http://www.ietf.org/html.charters/ipsec-charter.html, November 2002.

[PKCS] RSALaboratories, [www] http://www.rsa.com, November 2002.

[CSP] Microsoft Home Page [www] http://www.microsoft.com/msdn, November 2002.

[SSH] SSH Home Page [www] http://www.ssh.org. November 2002.

ANNEX I. PKIV6 (PUBLIC KEY INFRASTRUCTURE WITH IPV6 SUPPORT) TESTS

10.1 Introduction

This document describes the performance of the certification services offered by the Public Key Infrastructure designed and developed by the Department of Information and Communications Engineering of the University of Murcia (https://pki.umu.euro6ix.org) and the tests performed with the certificates obtained. These tests have been performed on several platforms for checking the reliability of the system.

10.2 Services

The services offered by this PKI have been acceded and tested from different browsers and mail client applications:

- IExplorer 5.5 and Outlook Express 5.5 on Windows 2000.
- Netscape 6.2.3 on Linux.
- Mozilla 1.0.1 on Linux.

The advances services like SCEP (Simple Certificate Enrollment Protocol) have not been tested.

10.2.1 IPv4/IPv6 Access

The access to the tested services has been realized over IPv6. It means that the communication between all entities involved in the services provision use IPv6 as communication protocol.

On the other hand, the tests using the issued certificates have been performed using IPv4 and IPv6 when possible.

10.2.2 Requesting a New Certificate

10.2.2.1 Explorer 5.5

When we want to request a new certificate, a single form is showed for requesting information to issue the certificate:

- Full Name.
- E-Mail Address.
- Organizative Unit.

and some contact information.

Besides the request data and the contact information, we filled the advanced form where we selected the "*Microsoft Base CryptoService Provider*" as CSP (Cryptographic Service Provider) instead of the default option. The remaining fields of the advanced form maintained the default options.

Once we have sent the request form, the system informed us about our certificate request number. After informing to the administrator of the PKI, we have received an e-mail digitally signed from him with information about the serial number of the certificate (in HEX format) and how to retrieve it.

For retrieving the certificate we must search for it and import it from this search page. The certificate can be imported as a personal certificate, if the associated private key is possessed, or as another person certificate.

The certificate will be loaded into the certificates storage of our web browser, in the following route:

Tools → Internet Options → Contents → Certificates

10.2.2.2 Mozilla 1.0.1

This browser only shows the single request form in which we specify personal data and contact information, besides the value of two certificate related parameters:

- Certificate extensions: SSL Client.
- Key length: 2048 (High Grade).

In both cases the default values are chosen.

The process to request a new certificate is the same as described in the case of Explorer 5.5.

The certificates storage where the certificate will be loaded is in the following route for this browser:

Edit → Preferences → Privacy & Security → Certificates

10.2.3 Searching for a Certificate

10.2.3.1 Explorer 5.5

If you want to check the data of one of your certificates you can easily search for it by clicking on this option of the left frame of the web site.

The searching is based on a form that presents two criteria for doing this:

- The serial number of the certificate.
- The e-mail address filled in the certificate request form.

You can select one of them, and the search results are the same independently of the chosen option. The data showed are:

- Certificate content.
- Certificate fingerprint.
- Cryptographic Service Provider (CSP).
- Export a certificate.

- Import this certificate.

When you want to import a certificate you must previously search for it the same way, and import it from the result page.

10.2.3.2 Mozilla 1.0.1

This service works the same way in Mozilla 1.0.1, the searching request form as well as the search results, request and show the same information.

10.2.4 Renewing a Certificate

The certificates issued have a validity period according to the system's policy. The certificates that we have obtained have a validity period of one year.

If the renewal request is accepted, the system's policy will automatically set a new validity period. The certificates that we have renewed extended its validity period in one year.

A revoked certificate can not be renewed.

10.2.4.1 Explorer 5.5

For renewing a certificate we only have to send a renewal request. If you have more than one personal certificate in the storage, the certificate we are using to accede to the PKI web site will be renewed

Once we have sent the renewal request, a signed e-mail from the PKI administrator will be received telling how to retrieve it, since for this new expiration date to be active we have to import the certificate again.

10.2.4.2 Mozilla 1.0.1

The process to renew a certificate is the same as described in the case of Explorer 5.5.

10.2.5 Revoking a Certificate

10.2.5.1 Explorer 5.5

For revoking a certificate we only have to send a revocation request previously selecting the reason from a list.

Once we have revoked the certificate we will not be able to use it, and the certificate will be issued in the CRL (Certificate Revocation List). This list consists of all the certificates that were revoked.

10.2.5.2 Mozilla 1.0.1

The revoking process is as commented for Explorer 5.5.

10.2.6 Search CA certificate and CRLs

The CA (Certification Authority) certificate will be used by the browser to check the trusting of the certificates issued by this CA.

The CRL (Certificate Revocation List) will be used by the browser to check if a certificate has been revoked before its expiration date.

10.2.6.1 Explorer 5.5

These two entitites can be easily obtained by means of sending a request from the same page.

10.2.6.2 Mozilla 1.0.1

We can get the CA certificate and the CRL the same way as commented for Explorer 5.5, but they both present some differences.

In the case of the CA certificate, once we have imported it to our certificates storage we have to edit it for enabling the three options that will allow us to use the certificates issued by this CA. These three options are:

- This certificate can identify web sites.
- This certificate can identify mail users.
- This certificate can identify software makers.

In the case of the CRL, we can choose to update it after a selected period of time.

10.3 Using Certificates

When you have obtained a certificate it is stored in the certificates storage, that shows your personal certificates, the certificates of other persons and the certificates issuers. Depending on the browser you are working with, this storage is acceding from differents routes.

Image 1 shows the certificates storage of Explorer 5.5.

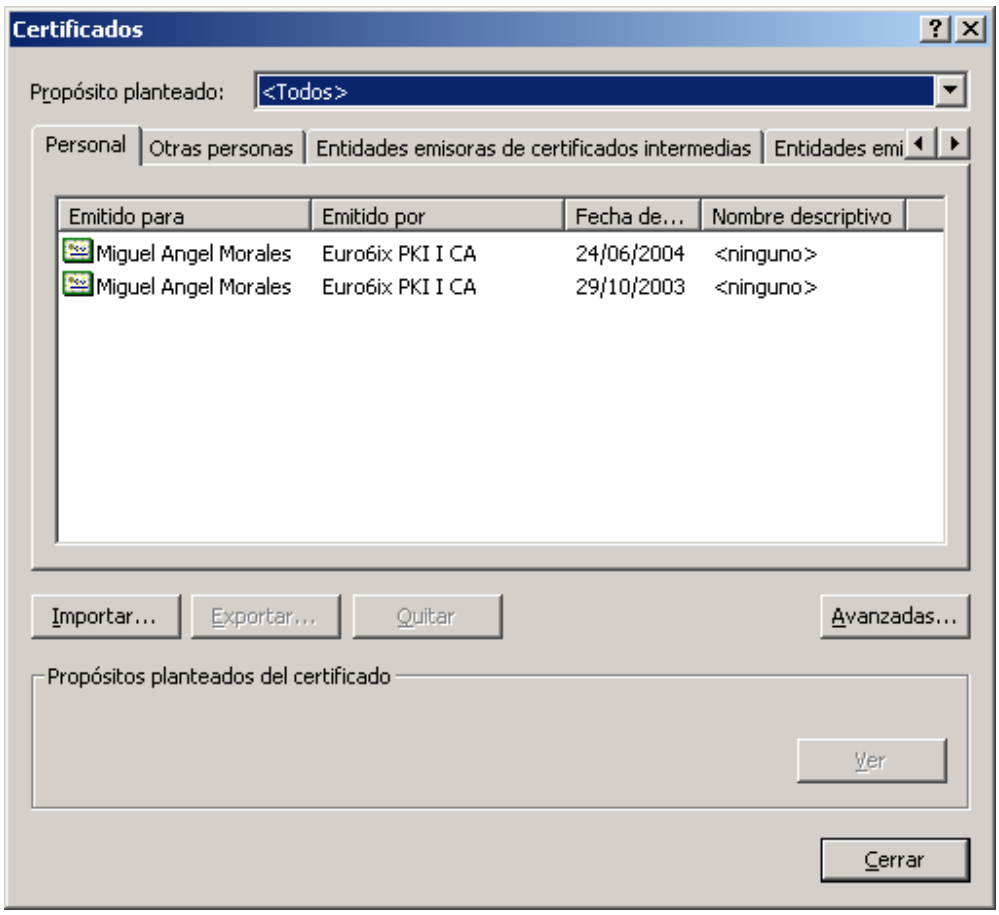

Figure 10-1. Certificates storage

If you are interested in the overall information about the certificate you simply select it and choose the '***See***' option, then you will be able to see something like figure 2.

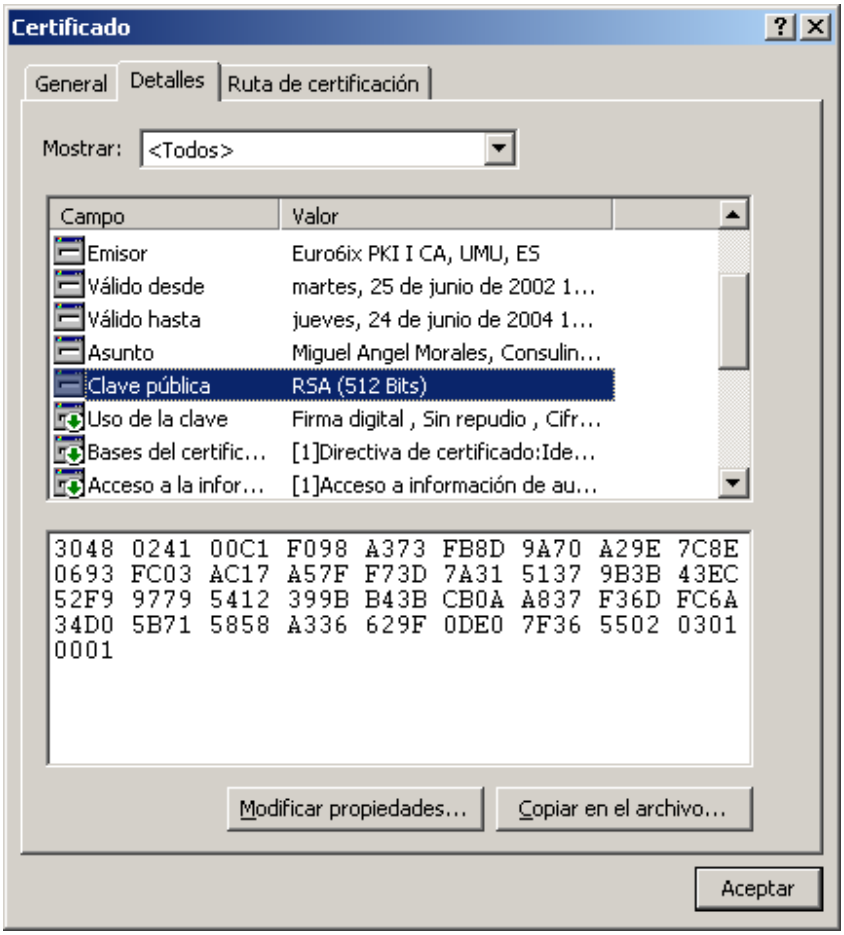

Figure 10-2. Certificate information

10.3.1 Browser Tests

For acceding to web sites by https protocol, a certificate must be used. When we accede to a secure site, an information message window like figure 3 is shown:

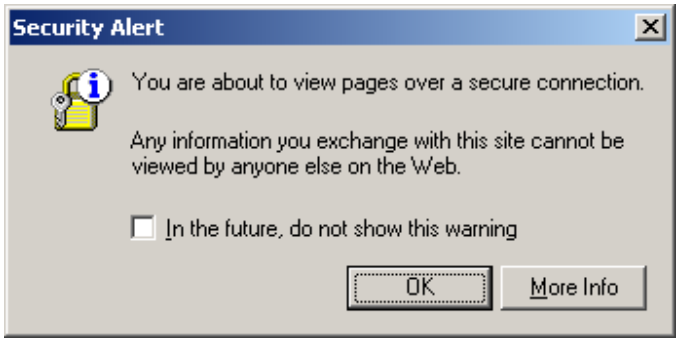

Figure 10-3. Access information message window

If you only have one certificate in the certificates storage, this one will be used for the access, but if you have more than one, a window will appear for choosing one of them. Figure 4 illustrates this event.

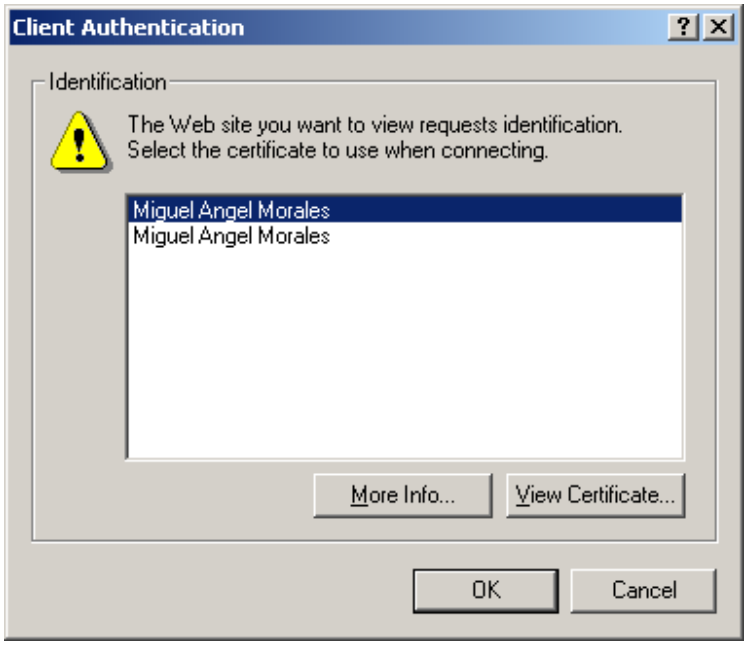

Figure 10-4. Certificate request window

Once you have chosen a certificate there will be an information exchange between the server and the client for checking the certificate used. If an error happened, a security alert window informs of that:

Figure 10-5. Valid certificate window

When leaving a secure site, another information window is showed:

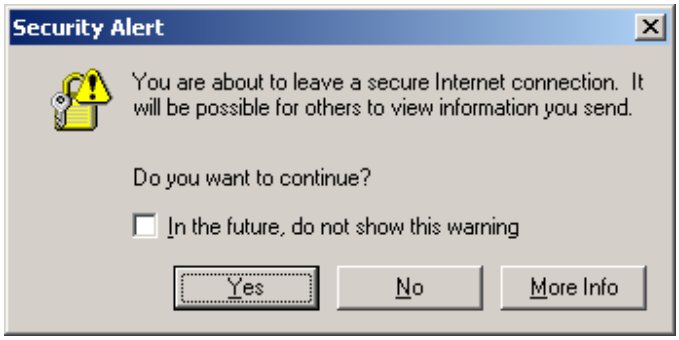

Figure 10-6. Leave information message window

10.3.2 Email Tests

The mail test consists of three main kinds of tests:

- Signed tests.
- Code tests.
- Signed and code tests.

10.3.2.1 Signed Tests

First of all we associate the certificate to an e-mail account, by this we can sign the e-mails sent from this account. When you sign an e-mail you are aggregating to the message your digital sign and your public key, what is called certificate.

Once we do this we will be able to send signed messages to everyone.

10.3.2.2 Coded Tests

When you code an e-mail you use the public key of the person who is the destination of the message to code it and this person uses his private key to decode it, and can verify your identity by your digital sign, so the sender and the receiver must have the certificate of each other.

The following table resumes sign and/or code tests performed.

Test	Source Platform / Mail Client	Destination Platform / Mail Client	Result
Sign	Windows 2000 / Outlook 5.5	Windows 2000 / Outlook 5.5	✓
Code	Windows 2000 / Outlook 5.5	Windows 2000 / Outlook 5.5	✓
Sign & Code	Windows 2000 / Outlook 5.5	Windows 2000 / Outlook 5.5	✓
Sign	Linux 2.4.18-3 / Mozilla 1.0.1	Linux 2.4.18-3 / Mozilla 1.0.1	✓
Code	Linux 2.4.18-3 / Mozilla 1.0.1	Linux 2.4.18-3 / Mozilla 1.0.1	✓
Sign & Code	Linux 2.4.18-3 / Mozilla 1.0.1	Linux 2.4.18-3 / Mozilla 1.0.1	✓
Sign	Linux 2.4.18-3 / Mozilla 1.0.1	Windows 2000 / Outlook 5.5	✓
Code	Linux 2.4.18-3 / Mozilla 1.0.1	Windows 2000 / Outlook 5.5	✗
Sign & Code	Linux 2.4.18-3 / Mozilla 1.0.1	Windows 2000 / Outlook 5.5	✗

Figure 10-7. Mail tests table

Linux machines have a Red Hat Linux 7.3 2.96-110 distribution.

Last two tests failed due to the following reason:

Mail clients uses destination certificates for coding e-mails, and since Outlook 5.5 and upper versions look for theses certificates in the storage, the mail client of Mozilla 1.0.1 force to choose the certificate for signing and coding e-mails, only showing personal certificates for selecting, so we were not be able to select the certificate of the destination mail account.

10.4 Conclusions

During the tests we have checked that the footers added to e-mails by mail servers are not compatible with the coding concept, since the footer is added when the e-mail has been coded, so when the e-mail arrives to the destination its content was modified by the mail server and the original content of the e-mail can not be sured.

Due to this event, the tests were performed using mail accounts the server does not modify adding footers.

This fact and failed tests do not mean the PKI does not work correctly since the certificates issued have always worked as waiting.

Mozilla 1.0.1 and Netscape 6.2.3 tests were performed in parallel but this last resulted to be less stable, so we will work with Netscape 7 in the future.