

The EuroPKI Experience

Antonio Lioy, Marius Marian, Natalia Moltchanova, and Massimiliano Pala

Politecnico di Torino, Dip. di Automatica e Informatica
Corso Duca degli Abruzzi 24, Torino (Italy)
security@polito.it

Abstract. This paper discusses the technical and management experience gained in the day-by-day operation of the EuroPKI infrastructure. First the context where EuroPKI was born is explained, along with its certification philosophy. Then common certification practices are discussed, along with description of the services and applications offered by the EuroPKI partners. User-reported problems are also listed and discussed in order to identify the issues that hamper large scale adoption of public-key certificates. The article closes with an overview of the EuroPKI activity plans and perspective.

1 Introduction

EuroPKI is a spontaneous aggregation of partners that believe in the value of public-key certificates and their use for network, application and document security. EuroPKI has its roots in the ICE-TEL and ICE-CAR projects, funded by the European Commission (EC) to promote the development of PKI-based European security technology to protect open networks and advanced network services (such as e-government, e-commerce or e-healthcare).

In July 1996 the ICE-TEL project began operation of its experimental PKI in 10 European countries. This is probably the first example of a running transnational European PKI. Initially, it was based on X.509v1 certificates that were later replaced by X.509v3 certificates to overcome the well known limitations of the v1 format (e.g. lack of expressivity, poor CRL management capabilities). The ICE-CAR project continued on this track, with more emphasis on support for real-life applications.

In January 2000 the ICE-CAR partners decided to create EuroPKI in order to broaden the scope of the infrastructure and to lay the foundation for its autonomous life beyond the end of the project. At the same time, the management of the Root Certification Authority (initially run by UNI-C in Denmark) was assigned to the Politecnico di Torino, who offered to maintain it at least until 2010.

After the end of ICE-CAR, EuroPKI has provided services to other national and international initiatives. Among them, the NASTEC EC-funded project used EuroPKI to promote secure applications in the newly associated states (NAS) to the European Union. Within this frame, two new national CAs were set up in Romania and Poland.

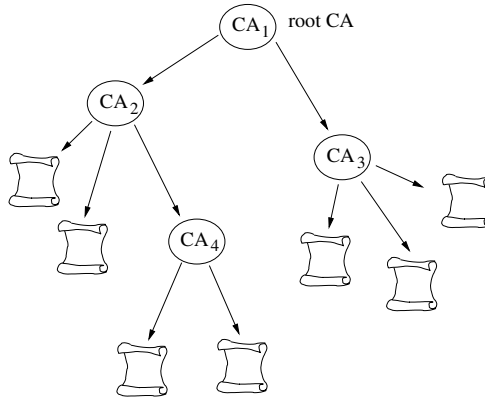


Fig. 1. Hierarchical trust model.

2 The EuroPKI certification model

The organization of a PKI reflects the trust model required by its constituency. Three primary models are used: hierarchical, cross-certification, and bridge.

Hierarchical trust (Figure 1) is the most common PKI model: trust is established as a tree structure that flows from top to bottom. At the top of the hierarchy is the root or top-level CA (TLCA): it directly certifies subordinate CAs that in turn provide certification services to their users or to other sub-CAs. This model permits to delegate trust together with CA operations to subordinate authorities. Also, the path construction procedure is very simple (a single path exists from any end entity up to the TLCA). The main disadvantages are the presence of a single point of failure (i.e. the TLCA) and the political issues related to subordination between parties.

In the cross-certification model (Figure 2), there is no subordination: two CAs cross-certify each other if they mutually agree to trust each other's certificates as if they had been issued by themselves. In this model there is no single point of failure, and the CAs are able to act fairly autonomously without being bound by policy delegated from a parent CA. Although many vendors have im-

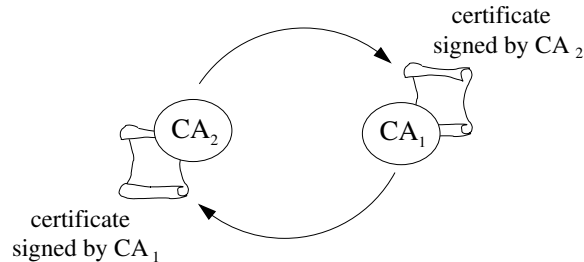


Fig. 2. Cross-certification trust model.

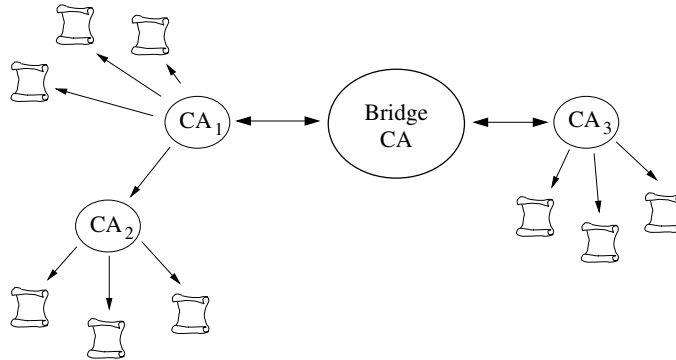


Fig. 3. Bridge trust model.

plemented cross-certification in their PKI management products and the IETF has included CA cross-certification in its Certificate Management Protocol [1], cross-certification is still not well supported by real-life applications.

The Bridge trust model (Figure 3) is similar to the cross-certification one in that there is no single Root CA [2] rather there is one CA - the Bridge CA (BCA) - that acts as a facilitator to interconnect other CAs. A BCA typically does not issue certificates to end entities, but it is used as a hub to interconnect the spokes that can be individual CAs, hierarchical or cross-certified PKIs. With this model, each member needs only to maintain a single cross-certification with the BCA and then it is automatically able to build certification paths across all spokes.

Trust is built into the applications by embedding the certificates of the trust anchors. Usually this takes the form of a *trust list*, that is the list of all CAs directly trusted by the application. Other CAs - in order to be trusted - must have a relation to one CA of the list and the relation type (i.e. hierarchical, cross-certification, bridge) must be understood and managed by the application. Trust lists are of common use in major off-the-shelf applications and provide a very simple solution to the trust management problem for the average user, but they are criticized because the criteria for insertion of a CA in the list are often based more on a commercial rather than a security analysis.

As EuroPKI wants to provide support for every-day user operations, a hierarchical trust model was adopted because this is the only model widely supported by currently available applications. Moreover, it requires only one operation (i.e. addition of the TLCA) to embed trust in the end-user applications, given that the partners didn't want to pay the fees needed to enter commercial agreements with the major application providers. At March 2004, Politecnico di Torino hosts the EuroPKI TLCA. As the number of affiliated national CAs varied since the start of the project, the TLCA has issued so far 18 certificates for 8 different national CAs.

One important feature of EuroPKI is the guarantee of a common ground where all partner operate according to a pre-established set of rules. These rules are specified in the EuroPKI Certification Policy (CP) [3] that establishes CA management rules and the applicability of issued certificates. A well-written CP may help end-users in deciding whether a certificate, and the binding therein, is sufficiently trustworthy for a specific purpose.

The first version of the EuroPKI Certification Policy (CP) was published in October 2000. The policy follows pedantically, section-by-section, the structure of RFC-2527 [4] to allow an easy comparison with this standard. Where a specific section of this RFC was not applicable to EuroPKI, an explicit note was made in the document, but that particular section was not skipped.

The EuroPKI policy successfully stimulated the partners to address issues in their own policies before submitting their versions to the hierarchy. Moreover, the EuroPKI policy was used as a starting point by other organisations, including even some not affiliated to EuroPKI (e.g. the CINECA Grid), when writing their own policy.

3 Common practices in EuroPKI

An internal survey was conducted at the end of 2003 to better understand the common practices followed by the partners. Gathered data are reported and commented here.

3.1 Certificate data and status distribution

Every partner CA make certificates available via HTTP (100%) and the majority supports also LDAP (54%). Every CA (100%) generates CRL that are made available via HTTP (100%) and sometimes also via LDAP (28%). No FTP or SCEP support was reported for certificate and CRL distribution. 45% of the partners support OCSP. No use of delta CRL [5] was reported, probably due to the small number of certificates revoked so far.

3.2 Certificate and CRL profile

The survey evaluated also the type and number of extensions used within certificate profiles. The results are summarised in Table 1. The certificate profiles are very similar to each other. The *CRL Distribution Point* extension is widely adopted while the *Authority Information Access* is reported to be used only by some organisations. This is reasonable since not all CAs support the OCSP service.

As will be explained in more detail in section 5.1, a wide adoption of the *Authority Information Access* and *Subject Information Access* extensions should be encouraged as it could help the certificate validation process and the retrieval of certificates.

Table 1. Certificate extensions usage.

Extension Name	Usage (%)
CRL Distribution Point (CDP)	73
Certificate Policy Identifier	91
Authority Information Access (AIA)	54
Subject Alternative Name	91

The survey compared also CRL profiles in terms of CRL version and frequency of publication. It turned out that all CAs publish CRLs at least once a month and almost every CA (82%) uses the v1 CRL format. This is due to the need to keep compatibility with old software (e.g. Netscape Communicator) that does not support the v2 CRL format.

3.3 CA management major issues

CA management is not an easy task, therefore many issues were reported concerning this subject. Three major problem areas have been identified:

- certificate profile management complexity and lack of flexibility
- character encoding support
- lack of needed features in CA management software

The first problem could probably explain why the certificate profiles are so similar across different organisations. In fact, poor understanding of certificate extensions could be the main cause for profile similarity.

3.4 User related issues

Users have reported several issues that fall mainly in two broad categories: PKI understanding and mobility problems.

The first class of issues is tied to the lack of knowledge about PKI fundamentals and digital signature technology. As the user interfaces of PKI-enabled applications aren't simple enough and foolproof, lack of PKI knowledge creates a usability problem for the average user. This is not a problem that can be solved by the certificate provider, but rather a call for better certificate handling in applications. Within EuroPKI, several partners have run seminars and prepared user-level documentation to help in using certificates within common applications, but a lot of work in this area is still needed.

The second class of problems is related to user mobility. Users often need to use digital certificates on different workstations and no easy solution is supported by existing applications. Moreover, cryptographic hardware devices (e.g. smart cards or USB tokens) are not yet widely deployed because specific drivers and hooks are needed, both at the OS and application levels, due to scarce support for standard-based cross-platform solutions such as *PKCS#11* [8].

Nevertheless, possible workarounds do exist. For example, applications could be configured to retrieve user credentials and trust anchors from a *PKCS#12* formatted file [7] that might be stored on a removable device like a floppy disk or a USB memory token. As all the user data is carried within this file (i.e. user's certificate, private key and other PKI data), no installation of additional software or extra configuration would be required. Also, the adoption of a more standardised approach for the integration of crypto-devices should be considered by applications and crypto-hardware vendors since this would help users in avoiding security problems (e.g. malicious code seeking for memory-stored crypto keys) and key-recovery issues (e.g. after a system crash).

3.5 Most valuable PKI-enabled applications

EuroPKI partners find that S/MIME secure mail is the most popular application. The second place is shared by access authentication (e.g. via SSL/TLS client authentication) and electronic document signature and encryption.

4 EuroPKI services and applications

Since the real value of a PKI is not in the certificate itself but in its application, several EuroPKI partners have developed certificate-based applications and provided application-oriented services, such as on-line certificate status verification (via OCSP) and Time Stamping (via TSP). To simplify the enrolment procedures, Registration Authorities have been studied and deployed. Moreover, different PKI-based applications have found a good test ground in this framework; an example of a successful one, WebSign, is described in section 4.4.

4.1 The centralised EuroPKI OCSP responder

The *online certificate status protocol* (OCSP) - defined in RFC-2560 [6] - is used to check on-line the status of a single certificate, so avoiding the burden associated to CRL download and analysis. TORSEC¹ developed back in the year 2000 an OCSP responder to be used within EuroPKI.

The policy behind the EuroPKI hierarchy favoured the scenario with a centralised OCSP responder that provides its service on behalf of all the CAs in the hierarchy. To this aim, the TLCA issues to the centralised OCSP responder a special certificate that contains two particular extensions:

- ***extendedKeyUsage*** with the *id-kp-OCSPSigning* object identifier. In this way the TLCA delegates the authority of signing OCSP responses to the owner of this certificate.
- ***id-pkix-ocsp-nocheck*** with a NULL value to state that the OCSP responder's certificate can be trusted across its entire validity period. The usage of this extension suggests to constrain the lifetime of the certificate: as this

¹ The security group of Politecnico di Torino - <http://security.polito.it>

extension allows the client to skip validation of the responder's certificate, to minimise the risks related to key compromises and their effects on the PKI relying parties, the lifetime of the server's certificate was restricted to a maximum of 4 months.

With this certificate profile, the OCSP server acts as a *delegated* responder for the EuroPKI TLCA, and as a *trusted* responder for all the other CAs present in the hierarchy. In literature, this mixture of the two traditional OCSP trust models is sometimes called the *family* model.

The EuroPKI OCSP responder uses CRLs as its source of revocation information, to preserve consistency between statuses obtained via CRL and OCSP. To always have the most accurate and up-to-date revocation information, the responder periodically looks for a new CRL on each CA in the hierarchy and eventually downloads it. CA administrators can also explicitly trigger CRL retrieval whenever a new CRL is issued. Each time a new CRL is downloaded, the OCSP responder performs a restart to load the new revocation data.

To avoid replay attacks, the EuroPKI responder is not using pre-produced answers: it always signs the OCSP responses on the fly. However this poses performance problems. The main bottleneck singled out during the server's development and testing is represented by the cryptographic capabilities of the hosting system's CPU. The throughput of the responder is always upper bounded by the maximum number of signatures per second that the hosting system is able to perform. Consequently, a considerable part of the development effort was devoted to minimise the overhead of the OCSP protocol itself. The tests run on various hardware platforms demonstrate that the server's performance is within 80-90% (in terms of requests per seconds served) of the maximum cryptographic performance of the server's CPU (in terms of signatures per seconds made).

4.2 The Registration Authority service

Under the EuroPKI policy an out-of-band ID authentication is required before a certification request is approved and the corresponding certificate is issued.

In most PKI environments, the request authentication may be performed by a Registration Authority (RA). There can be several scenarios. For example one organisation may outsource the certification service from an external vendor but still internally perform the RA function. Another scenario could involve the presence of many RAs disseminated over a wide area to maintain close contacts with the users. In all these cases, the CA would require proper authentication of the RA operator before issuing a certificate.

There are different solutions to the problem of RA operator authentication. For example, the OpenCA project implements web-based signed forms to approve users' request via common browsers (i.e. Internet Explorer, Mozilla or Netscape Communicator). This approach uses existing technologies - such as Javascript and SSL/TLS client authentication - to perform on-line RA operations. This approach has some limits, though. First of all, web-based form signing

is not supported by every browser and anyway it is not standard, so that different scripts are required depending on the target browser. Additionally, all the RA operations must be performed on-line.

To avoid these limitations, TORSEC developed an RA tool to connect securely to the CA front end and perform operations request approval or rejection. This software, based on the OpenSSL crypto library, has a graphic Win32 client for the RA operator and a Unix gateway to the CA management software. Operator identification is possible via software or hardware cryptographic credentials: the client supports *PKCS#12* files as well as smart-cards and USB crypto tokens via a *PKCS#11* interface to the OpenSSL Engine extension.

The RA operator connects to the server over an SSL channel that requires certificate-based client authentication. Access to a specific queue of pending certification requests is then granted by checking the operator identity against a signed Access Control List (ACL). Multiple ACLs can be stored on the same server and they can be related to one or more CA. This allows the RA server to be accessed by RAs affiliated to different CAs at the same time. Once authenticated, the operator can download the pending requests, sign the approved ones and send them back to the RA server. The requests are then queued on the server and wait to be exported to the corresponding CA for certificate issuing. As different RAs can access the same server, multiple operators can exist in different locations thus allowing third parties to act as Registration Authorities on behalf of one CA. A schema of the RA data flow is shown in Figure 4.

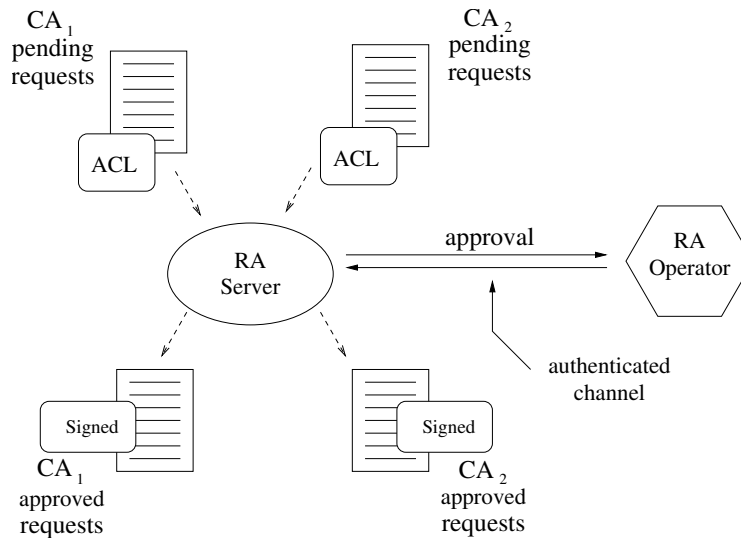


Fig. 4. RA client-server data flow.

4.3 The EuroPKI Time Stamping service

A time stamp is an electronic seal that includes a time indication. In practice it's a digital signature over a submitted digest, a time indication and other information. The RFC-3161 [9] defines the Time Stamping Protocol (TSP) to be used for requests and responses between the entities involved: the requester or client, and the Time Stamping Authority (TSA) or server.

Based on software developed by the TORSEC group, EuroPKI has established a TSA connected to a NTP stratum-1 server. The TSA server is compliant with RFC-3161 but supports only raw TCP access. For security reasons, client access can be restricted by accepting only signed requests or SSL connections with client authentication.

The TSA certificate has some peculiarities. Unlike the OCSP responder, there is no problem with key compromise: if it happens, the TSA certificate can be revoked and hence no special restriction must be posed to the certificate lifetime. However, to be used for time-stamping, the certificate contains the *extended-KeyUsage* extension with only the *timeStamping* value, marked as critical.

4.4 WebSign, a successful PKI-based application

WebSign is a client-server platform for on-line document preparation, submission, signing, storage and processing. This platform was developed by SETCCE² for use in various business, academic and governmental scenarios, where a transition from paper based records to electronic documents enhances the productivity and reduces the costs. The platform is based on XML documents with XML-signatures and advanced security mechanisms, to provide a high security.

The service has been put into production at SiOL, the biggest Slovenian Internet operator. The platform developed will foster the use of electronic contracts and digital signatures in business environments. Through WebSign, users are able to manage subscription and other Internet provision related services. The WebSign service has been introduced to the public in September 2003 and in the first 6 months the application client has been downloaded from the provider site about 12,000 times. The WebSign users use Qualified Certificates (according to the e-signature European directive) issued by four Slovenian certification service providers.

5 Known issues

5.1 Repositories and data retrieval

CAs are required to publish certificates and CRLs in public repositories. Although this could not seem a serious problem, general and simple availability

² Security Technology Competence Centre - <http://www.setcce.org>; funded by Slovenian Telecom, Jozef Stefan Institute, University of Ljubljana and Infotehna, SETCCE is the administrator of the EuroPKI Slovenian branch

of PKI data is still an unresolved issue. This is related to two different aspects: lack of pointers to the available data and the variety of access protocols.

The problem is very evident when considering people, not being part of a partner organisation, who look for one or more issued certificates. Without focusing on trust related issues, a feasible solution to correctly provide pointers to published data is an extensive usage of the *Authority Information Access* (AIA) and *Subject Information Access* (SIA) extensions. The former can provide information on the issuer of the certificate while the latter carries information (inside CA certificates) about services offered. The *Subject Information Access* extension can carry an URI to point to certificate repositories and Time Stamping services. Hence this extension allows to access services by several different protocols (e.g. http, ftp, ldap or smtp).

Within EuroPKI, public data is mostly available via HTTP. Other protocols (e.g. LDAP) are seldom used or not advertised outside the issuer's organisation. The AIA and SIA extensions are almost never used to point to data sources.

5.2 Multiple e-mail addresses support

One of the most valuable PKI applications is protection of e-mail messages. Although RFC-3280 requires specification of the e-mail address in the *subjectAltName* extension, some CAs still encode this information in the Distinguished Name (DN) by using the *Email* field. This has an obvious disadvantage when support for more than a single address in one certificate is required: multiple *Email* fields could raise compatibility issues with widely adopted e-mail clients.

To test application support for multiple e-mail addresses in the same certificate, we issued certificates under two different profiles. In profile A two different e-mail addresses were coded in the *subjectAltName* extension, while in profile B they were inserted by using multiple *Email* fields in the subject DN. Table 2 shows the test results with popular S/MIME e-mail clients. Results confirm that the support for multiple e-mail addresses can be achieved only when using the *subjectAltName* extension. It should be noted that some organisations encode multiple e-mail addresses by using simultaneously the two described methods. Although this should be avoided, it is reported to work sometimes.

Table 2. E-mail client tests.

E-Mail Client	Profile A	Profile B
Netscape 4.7	Supported	Not Supported
Netscape 6	Supported	Not Supported
Netscape 7.1	Supported	Not Supported
Mozilla 1.5	Supported	Not Supported
Thunderbird 0.6	Supported	Not Supported
Microsoft Outlook 6	Not Supported	Not Supported

5.3 Non US-ASCII character sets

When eastern European countries entered EuroPKI, the problem of non US-ASCII characters in certificates had to be faced. In RFC-2277 [10] Alvestrand suggests the need to clarify whether the strings used in a protocol are subject to internationalisation or not. Although this RFC does not mandate for a policy on name internationalisation, it requires that all protocols be able to use the UTF-8 character set. This consists of the ISO 10646 [11] character set combined with the UTF-8 [12] character encoding scheme. The RFC-3280 promotes the adoption of the UTF8String encoding type to allow for special characters to be properly stored: conforming CA's must encode attributes of DirectoryString type as UTF-8 after December 31, 2003. While this can be more or less easily achieved when issuing certificates, the real problem lies on the application side. Several software packages still do not support the UTF8String type properly. We list here our findings on the behaviour of widely used applications.

Open issues are present in some versions of MS Windows. When a certificate is displayed by an application running under Windows-NT4, Windows98 or Windows95 and the subject field of the certificate contains UTF-8 characters, the data is not correctly visualised. According to Microsoft Knowledge Base Article 824197 the problem source is in the operating system itself and no fix is planned. Hence full UTF-8 support for X.509 certificates is currently available only in Windows ME, Windows 2000, Windows XP, and Windows Server 2003.

The Netscape Communicator suite (i.e. Netscape Navigator and Netscape Mail) up to version 4.x - still in use in some environments - does not support UTF-8 encoded strings. This causes the application not to even import user certificates containing this kind of data. It is therefore impossible to use such certificates with this suite.

The Mozilla and Firebird browsers had a bug which produced some unreadable characters when displaying certificate details. This was due to the fact that non US-ASCII characters (encoded as BMPString) were displayed by using iso-8859-1 encoding. Although it was not possible to read the special characters, no other function was affected. Although a patch for this bug (#185167) is available at <http://bugzilla.mozilla.org>, it has not yet made its way to the stable distribution at April 2004.

The Opera suite version 7.23 correctly supports certificates with UTF-8 encoded strings both in certificate visualisation and usage. However Opera uses certificates only for SSL since it doesn't support S/MIME in its mail client.

5.4 Certificate renewal

The EuroPKI policy permits to renew an expired certificate. It is therefore possible to issue a new certificate for the same public key, with no need to regenerate a new key pair. However this creates a problem with Internet Explorer that cannot deal with this situation: the new certificate simply doesn't show up in the certificate management interface.

A solution to this problem is to import the renewed certificate together with the corresponding private key; this can be done by using a PKCS#12 file where these data are bundled together. It seems that IE needs, for each imported certificate, a corresponding private key, thus being not able to bind more than one certificate to a single private key.

5.5 Naming rules in Distinguished Names

The EuroPKI policy does not impose specific rules about the format of the Distinguished Name beside the need for names to be meaningful. However, due to law or administrative requirements this could not always be true.

As an example let us consider the case of one affiliated Italian CA that needed a special format in the Common Name (CN) for compliance with the Italian digital signature law [13]. The requested format is:

$$CN = \langle\langle surname \rangle\rangle / \langle\langle name \rangle\rangle / \langle\langle personal_tax_id \rangle\rangle / \langle\langle unique_id \rangle\rangle$$

This format of the CN carries data meaningful only to applications compliant with the Italian digital signature law. Moreover the DN must contain another non optional field: the *Description* one. It is composed by several sub-fields that carry data able to confuse a standard X.509 parser even more:

$$Description = \langle\langle C = \langle\langle surname \rangle\rangle / N = \langle\langle name \rangle\rangle / D = \langle\langle birth_date \rangle\rangle / [R = \langle\langle role \rangle\rangle] \rangle\rangle$$

The presence of slashes or commas can raise compatibility issues because these characters are mostly used as fields separators and some parsers could misinterpret these contents. This is an example of bad usage of certificates that break compatibility across applications.

To improve interoperability and lifetime of certificates, it would be better that specific data, where absolutely needed, be encoded by using the *otherName* type in the *subjectAltName* extension. To certify roles, the adoption of Attributes Certificates should also be considered.

6 Future plans

Besides maintaining and expanding EuroPKI, our future efforts are aimed to develop better PKI support for applications and systems.

A first area to be addressed is better certificate processing in complex cases. Deep hierarchies, bridge CA, multiple sources of revocation status (CRL, OCSP, indirect CRL, ...) require careful definition of procedures when building the certificate path up to a trusted root and verifying the status of all the certificates in the chain. An exact algorithm needs to be defined and implemented as a library to support applications.

Proper configuration of thousands of workstations to support PKI is a nightmare for a security administrator. In the same way, implementing PKI support into light devices (such as PDAs or handheld PCs) with reduced networking and

processing capabilities is very difficult. These two problems can be simultaneously solved by the concept of “lightweight PKI” where activities that are heavy or require centralised configuration, are delegated to a trusted PKI server. Along this line, the IETF PKIX working group has already defined the requirements for delegating the validation process to a dedicated server. Various protocols, including OCSP with extensions, SCVP [14] and DVCS [15] were proposed for communication among the client and the delegated server. We have already started to investigate the proposed protocols and our efforts are directed towards implementing the servers and integrating support for certificate validation in light devices (e.g. we are currently working with Windows CE handheld PCs).

On the PKI management side, our efforts are directed to support bulk registration and renewal. Large universities and public administrations with tens of thousands of users face severe problems related to management complexity and registration services costs. For example, let us consider a University that needs to issue 30,000 certificates: if the registration process requires five minutes for each certificate (an optimistic value!) the whole process would require 150,000 minutes or more than 14 man/month! And this is without counting the time needed to solve technical problems or correct human errors. Therefore research and development of new tools to support bulk registration and approval is going on, to introduce a certain degree of automation in the certification process. Our current approach is to perform automatic data retrieval from existing data bases and to re-engineer the standard registration services.

7 Conclusions

Through its four years history, EuroPKI demonstrated its usefulness in supporting the security needs of end-users. Additionally, it has helped partner organizations to better understand the technical, policy and management issues involved in setting up and maintaining an operative PKI.

Experience has shown that the problems on the certification authority side can always be solved: technical standards do exist, products implement them and most of the times it is just a matter of proper configuration and management.

The real problems lie on the application and end-user sides. Application developers have not security as a priority, have limited understanding of PKI and are not well supported by transparent and complete libraries.

8 Acknowledgements

The authors want to thank the people and organizations that contributed to creating and developing the EuroPKI infrastructure: the current partners (CSP, Ezitrust, IAIK, NASK, Politecnico di Torino, Politechnika Lodzka, Politehnica University of Bucharest, Provincia di Macerata, PWR Centre for Networking and Supercomputing, SETCCE, University of Modena and Reggio Emilia) as well as the past ones and every partner of the ICE-TEL, ICE-CAR and NASTEC projects.

The authors also thank the European Community for the support provided by funding the mentioned research projects, SECUDE³ for providing free of charge its CA management suite to the EuroPKI TLCA, and the open-source community for many tools that we use in our everyday operations.

A special thank to those people that back in the '90s have introduced us to the wonderful field of PKI: David Chadwick, Borja Jerman Blazic, Steve Kille, Peter Kirstein, Sead Muftic and Wolfgang Schneider.

References

1. C.Adams, S.Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC-2510, March 1999
2. W.T.Polk, N.E.Hastings, "Bridge Certification Authorities: Connecting B2B Public Key Infrastructures", NIST, September 2000
3. "EuroPKI Certificate Policy - Version 1.1", EuroPKI website, <http://www.europki.org>
4. S.Chokhani, W.Ford, "Certificate Policy and Certification Practices Framework", RFC-2527, March 1999
5. R.Housley, W.Polk, W.Ford, D.Solo, "Certificate and Certificate Revocation List (CRL) Profile", RFC-3280, 2002
6. M.Myers, R.Ankney, A.Malpani, S.Galperin, C.Adams, "Online Certificate Status Protocol - OCSP", RFC-2560, June 1999
7. RSA Laboratories: "PKCS#12: Personal Information Exchange Syntax Standard", Version 1.0, June 24, 1999
8. RSA Laboratories: "PKCS#11: Conformance Profile Specification", Version 2.11, October 1, 2000
9. C.Adams, P.Cain, D.Pinkas, R.Zuccherato: "Time-Stamp Protocol (TSP)", RFC-3161, August 2001
10. H.Alvestrand: "IETF Policy on Character Sets and Languages", RFC-2277, January 1998
11. ISO/IEC: "Information Technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", May 1993, with amendments
12. F.Yergeau: "UTF-8, a transformation format of ISO 10646", RFC-2279, January 1998
13. AIPA: "CIRCOLARE 19 giugno 2000 n. AIPA/CR/24", Italian MIT Website, <http://www.innovazione.gov.it>, 2000
14. A.Malpani, R.Housley, T.Freeman: "Simple Certificate Validation Protocol (SCVP)", IETF Draft, PKIX Working Group, October 2003
15. C.Adams, P.Sylvester, M.Zolotarev, R.Zuccherato: "Data Validation and Certification Server Protocols", RFC-3039, February 2001
16. C.Weider, C.Preston, K.Simonsen, H.Alvestrand, R.Atkinson, M.Crispin, and P.Svanberg: "The Report of the IAB Character Set Workshop held 29 February - 1 March, 1996", RFC-2130, April 1997

³ <http://www.secude.com>