PKI-Based Trust Management in Inter-Domain Scenarios

Gabriel López Millán, Manuel Gil Pérez, Gregorio Martínez Pérez,
Antonio F. Gómez Skarmeta

Departamento de Ingeniería de la Información y las Comunicaciones
University of Murcia, 30071, Murcia, Spain
Email: {gabilm, mgilperez, gregorio, skarmeta}@um.es

Abstract. Hierarchical cross-certification fits well within large organizations that want their root CA to have direct control over all subordinate CAs. However, both Peer-to-Peer and Bridge CA cross-certification models suits better than the hierarchical one with organizations where a certain level of flexibility is needed to form and revoke trust relationships with other organizations as changing policy or business needs dictate. It seems that this second approach better fits the current and next-generation inter-domain networking models existing in both the wired and wireless Internet. In this context, this paper analyses some relevant inter-domain scenarios and derives the main requirements in terms of cross-certification from them. It then describes the design and lab implementation of a pan-European scenario which is based on a research network composed by a set of organizations that may have their own PKIs running, and that are interested to link with others in terms of certification services. It provides a complete design, implementation and performance analysis for this complex scenario, including a procedure and practical recommendations for building and validating certification paths.

Keywords: Trust Management, Inter-Domain PKI, Performance Evaluation, PKI Requirements.

1. Introduction and motivation

Some Public Key Infrastructures (PKIs) [1] are now starting to define and use certification structures based on advanced trust models (i.e. Peer-to-Peer and Bridge CA cross-certifications) rather than basic certification hierarchies. It is better serving the current Internet structure, which is defined as a set of interconnected networks acting as a single virtual network.

These certification models are based on the fact that each organization can manage its own PKI, and then to establish and revoke cross-links with others when necessary, for example, according to its internal policies or business needs. These links are based on cross-certification [2, 3] processes, that is, procedures undertaken by Certification Authorities (CA) to define trust relationships. When two CAs are cross-certified, they agree to trust and rely upon the digital certificates issued by them. It allows easy and scalable trust management between certified entities.

Two main cross-certification models are being currently used: Hierarchical cross-certification, which defines trust relationships between CAs inside the same administrative domain, and Peer-to-Peer cross-certification, which defines trust relationships between two autonomous (either stand-alone or hierarchical) CAs. A third alternative is the Bridge CA (BCA) model, representing a trustworthy independent node, which establishes trust relationships with several non-related CAs. Every CA

shares one (i.e. unidirectional relationship) or two (i.e. bidirectional relationship) cross-certificates with the BCA, thereby establishing a trust relationship between the CAs through this neutral point. A deeper analysis of those models can be found at [2].

In these three described cross-certification models, trust can be *initially* considered as *transitive*, that is, if A has a trust relationship with B and B with C, implies A has an *indirect* trust relationship with C. However, depending on the particular scenario and/or the involved organizations, this feature may or may not be desirable. Thus, before a cross-certificate is issued, all the requirements and constraints to deal with this issue (or similar ones) have to be negotiated and agreed between involved parties. Restrictions in cross-certification environments can be described using a well-known group of extensions, such as *Basic Constraints, Certificate Policies, Name and Policy Constraints* and *Policy Mapping* [3, 4]. The main objective of these extensions is to differentiate between CA and end entities certificates, to specify certification policies under those certificates have been issued, and to establish restrictions in the certification path for new issued certificates.

According to this, the main questions arising are: which trust model (or combination of them) should be deployed in a real inter-domain communication network? What is the best option in terms of performance? How can an entity (end user, application or device) determine whether the certificate provided by any other entity from a different organization can be trusted or not? And how the user response time is affected by the number of intermediate CAs taking part of the inter-domain trust infrastructure?

As there is not a common and agreed answer for all these questions, just some basic recommendations from the industry and the standardization bodies, we think that the provision of a practical experience related to the definition of a large-scale inter-domain scenario can be of interest for PKI designers and implementers. It can also help to promote a wider adoption of cross-certification trust models. This is the main motivation of this paper, where we describe the design, implementation and performance measurement of a cross-certification scenario. For this, we have taken the requirements from several scenarios including a real pan-European research communication network built during the Euro6IX IST European research project [5], and which was composed by several security domains willing to link securely their certification services.

This paper is structured as follows. Section 2 describes the inter-domain scenarios that will be considered throughout this paper. The main requirements for these scenarios in term of PKI services, certificate extensions, and certification path building and validation are provided in section 3. Later, section 4 presents the design of a lab testbed, which is then used in section 5 to validate the ideas presented in previous sections. Section 6 provides a discussion about the lessons learnt from our research work. Section 7 presents some related works. Finally, we conclude the paper with our remarks and some future directions derived from this research work.

2. Inter-domain scenarios

This section introduces three of the main current scenarios that demonstrate how the establishment of trust relationships between organizations may become a complex process.

2.1. Identity federations

This scenario is based on the definition of a trust relationship among service providers (*remotes organizations)* and identity providers (*homes organizations)*, in order to allow the exchange of end user credentials and related information among organizations. Examples of identity federations are InCommon [6] or SWITCH [7], for web services, and eduroam [8] for network services.

Although in federations composed by few participants it is not necessary to deploy a complex cross-certification system, there are others scenarios where the number of participant organizations makes difficult the management of single CA hierarchies. One clear example of this situation is eduroam, where more than 100 organizations, from 33 countries of three continents make use of the same network access service.

Identity federations like eduroam are already in production, giving network access service to thousands of users. Now, next steps head to the definition of collaboration among those federations; what is called *confederation*. For example, U.S. research and education community is working on a similar solution to eduroam for U.S. institutions [9]. It seems clear that these federations will end up establishing trust relationships to define a confederation. For example, a typical cross-certification scenario between a European PKI hierarchy and its USA counterpart could involve until six subordinate and root CAs between two belonging organizations.

2.2. e-business BCA

Nowadays, most of the e-business scenarios are focused on the establishment of trust relationships among companies and organizations around the world. We can find several organizations establishing trust relationships based on the Bridge CA model, in order to define common and neutral trusted entities. Some examples are the following:

- European Bridge-CA (EB-CA) [10] enables a secure communication channel between businesses and public authorities, including 35 members among the main banks, assurance, and telecommunication companies around Europe.
- Chinese Taipei BCA [11] allows interoperability among public and privates CAs, and defines a framework to enable certification services by bridging public root CAs, financial CAs and foreign CAs. This organization has issued more than 1.500.000 certificates, supporting more than 350 PKI services. It also supports four subordinates CAs and eleven CA companies.
- The last example is the FBCA (Federal Bridge Certification Authority) [12], which enables transitive trust among U.S. entities cross-certified with the FBCA. More than 20 organizations (CAs) are collaborating under the FBCA umbrella.

2.3. Telcos and service providers

Another important scenario where the establishment of complex trust relationships is becoming an important matter for security administrators is the one composed by network operators and service providers willing to collaborate to provide content distribution services to the end user. An example of this approach was the Euro6IX pan-European network.

Euro6IX [5] aimed to design and deploy a pan-European native IPv6 network, and to research on advanced and innovative network services and applications. Euro6IX deployed a communication network based on three main elements: IPv6 IX (Internet Exchanges) nodes, IPv6 networks, and end entities (final users, software processes and/or hardware devices) connected to these networks. The IX nodes, which include most of the security services, run at the research branches of the telecom operators taking part in the project, i.e. Telefonica, British Telecom, France Telecom, Deutsche Telekom, Telecom Italia and Portugal Telecom.

In this scenario, a Bridge CA model was implemented and required six cross-certification relationships between IXs CAs and BCA, while each IX defined its own single, hierarchical or cross-certificated infrastructure internally. A typical certification path between two organizations could involve five or six CAs, plus the corresponding cross-certificates between them.

3. Requirements of inter-domain trust management infrastructures

Scenarios described in section 2 suppose that every security domain or organization has a valid PKI running one or more certification authorities, which are governed by some kind of internal policies; thus, each organization will have to define an internal certification model (single, hierarchical or peer-to-peer cross-certification), and the mechanism to establish external security relationships.

In order to provide the required level of functionality and reliability to these three scenarios, each security domain has to implement a minimum set of certification services, building and validation algorithms, and issue public key certificates with some extensions requirements. These are presented and motivated in next subsections.

3.1. PKI service requirements

One of the main services that need to be offered to an entity (e.g. end user, device or software process) is the possibility to determine whether the certificate provided by any other entity can be trusted or not. This decision will be based on the existence of a valid certification path between the target certificate and a Trust Anchor [13].

The infrastructure has to ensure that the path can be built and validated in real-time and several services are necessary to be implemented by every organization involved in the certification path. This work supposes every PKI provides the following services:

- LDAP: directory services have to include, beside CAs and end user certificates, cross-certificates for each trust relationship with other domains, CRLs and ARLs.
- Validation Service (VS): according to [14], mandatory in every security domain to allow relying parties to delegate the validation process to a third trusted party. Protocols, like SCVP [15] or OCSP [16], can be used.
- Time Stamping Protocol (TSP): certifies timestamps, helps protocols like OCSP.

The first two services are critical in public key infrastructures in order to provide building and validation mechanisms to third trusted parties. Current solutions provide CRL-based validation mechanisms and some of them also offer advanced services like OCSP. However, neither CRL nor OCSP were designed to provide advanced

certification path building and validation, but simple certificate status request. Indeed, a more suitable protocol such as SCVP has to be deployed for these cross-certification scenarios. Thus, we define the use of SCVP to interact with the VS.

3.2. Certificate requirements

There are three main types of public key certificates that can be defined in the inter-domain scenarios: end entities, CA, and cross-certificates. The first ones are issued to end entities (users, devices or software applications). The second ones belong to every CA existing in the scenario; including root, subordinates and bridge CAs. Finally, cross-certificates are issued between CAs, where a typical bidirectional relationship needs two cross-certificates, named *forward* and *reverse* [2, 13]. These cross-certificates establish the constraints and policies [4] defined by the agreement between two CAs.

According to our research, testing, later feedback from Euro6IX, and the recommendations of the PKIX WG [4] and [17], seventeen extensions may appear in every certificate, although depending on the type of certificate each extension will be considered as mandatory (M), recommended (R) or optional (O). Table 1 summarizes the relevant certificate extensions.

- *AuthorityKeyIdentifier* and *SubjectKeyIdentifier* extensions help path building algorithms to select between signing certificates of the same entity.
- *KeyUsage* extension must be included in every CA certificate. When a certification path needs to be selected among several candidates, cross-certificates with *KeyUsage* and *BasicConstraints* with *cA* value set to true will have priority.
- *CertificatePolicies* is strongly recommended to be used in every certificate to know under which particular security policy has been issued.
- *PolicyMappings* extension uses the *CertificatePolicies* identifiers to map different certification policies defined by different domains. *PolicyMapping* is mainly used in the BCA↔CA cross-certificates establishing equivalence between policies.
- *SubjectAlternativeName* and *IssuerAlternativeName* extensions help a validation service to select CA certificates and make faster the path building process.
- *SubjectDirectoryAttributes* is an optional extension for including identification purposes of the subject.
- *BasicConstraints*. Every CA certificate must set the *Path Length* attribute depending on the internal structure. Cross-certificates issued by the Bridge CA must set *Path Length* to *optional* to avoid limiting the certification paths from it.
- *NameConstraints* extension should be defined in every cross-certificate. This extension, mainly used by the BCA, can be used both in a BCA→CA cross-certificate to ensure that a CA will always issue end entities and subordinate CAs certificates under a specific *Name*, and in a CA→BCA certificate to exclude possible authentication between external organizations that the CA does not trust.
- *PolicyConstraints*, used to control the policy in a trust chain, can be included inside the cross-certificates issued to or by the BCA. The CA→BCA may contain an *inhibitPolicyMapping* with value 1 to ensure that the *PolicyMapping* will be taken into account only between the CAs certified by the BCA.

- *ExtendedKeyUsage* appears only in end entity certificates for adding more purposes of use than the included ones in the *KeyUsage* extension.
- *CRLDistributionPoints* extension points out where CRLs/ARLs can be obtained, and should be included in every certificate to ensure validation processes.
- *InhibitAnyPolicy* restricts the number of non-self-issued certificates that may appear in a certification path until reaching another self-issued CA.
- *FreshestCRL* is an optional extension that points out how to retrieve delta CRLs.
- *AuthorityInfoAccess* extension is considered mandatory and must be included as an easy way to allow recovering the CA and cross-certificates. It is also used to provide both a contact point where OCSP queries can be done and the URL of the LDAP repository. *SubjectInfoAccess* is highly recommended to be used in every certificate, regardless its type.

Table 1: Certificate extension usage

X.509 Extension	CA Cross-Cert	Root CA	End Entity
AuthorityKeyIdentifier	M	O	R
SubjectKeyIdentifier	M	M	O
KeyUsage	M	M	R
CertificatePolicies	R	R	R
PolicyMappings	R	O	-
SubjectAlternativeName	O	O	O
IssuerAlternativeName	O	O	O
SubjectDirectoryAttributes	O	O	O
BasicConstraints	M	M	-
NameConstraints	M	O	-
PolicyConstraints	O	O	O
ExtendedKeyUsage	-	-	R
CRLDistributionPoints	R	R	R
InhibitAnyPolicy	O	O	-
FreshestCRL	O	O	O
AuthorityInfoAccess	M	O	M
SubjectInfoAccess	R	M	M

3.3. Discovery and validation of certification paths

Generally speaking, a certification path building algorithm can be seen as a *tree traversal* algorithm assigning weights or priorities to each tree branch or link in the decision-making process. Therefore, the idea in the discovery of certification paths, as published in [18] and used in this work, can be simplified to a tree search algorithm based on a *best path first*.

This algorithm builds the certification path from the target certificate to a recognized Trust Anchor, given by a *relying party*. This way of construction is known as forward direction [19]. Figure 1 depicts the main blocks of the designed and implemented algorithm. The complete workflow and a detailed description of this algorithm can be found in [18].

The VS providing this algorithm deploys a SCVP interface for relying parties. After verifying and processing the SCVP request (queried certificates, Trust Anchors, *wantBack* information, etc.) by the VS, the algorithm is called *recursively* taking the last certificate added to the candidate certification path as the current one (initially, this path is composed by the target certificate). As this certificate has not been treated yet, the algorithm tries to explore from this certificate whether there is a valid certification path to one of the defined Trust Anchors. This search is performed through different kinds of cross-certification models:

- *Hierarchical models*. If the certificate the algorithm is treating does not represent a root CA, the next link from which to continue investigating is its own issuer certificate (intra-domain search). Thus, the algorithm collects, from the corresponding LDAP repository (whose URI is defined in the *id-ad-caIssuers* attribute of the *AuthorityInfoAccess* extension), all needed information of the current certificate's issuer (CA certificate, CRLs, ARLs and a set of cross-certificate pairs) by executing only one LDAP request.
- *Non-hierarchical models*, such as Peer-to-Peer and Bridge CA. The algorithm checks whether the current CA certificate has got any relationship with other trusted domains through a cross-certification process (inter-domain search). Thus, the *forward* cross-certificates, which represent these relationships following a forward direction, are queued for being explored one by one. They symbolize each of the possible paths the algorithm can choose until reaching one of the defined Trust Anchors. The algorithm is recursively executed again with one of the above *forward* cross-certificates to continue searching through a new promising lead.

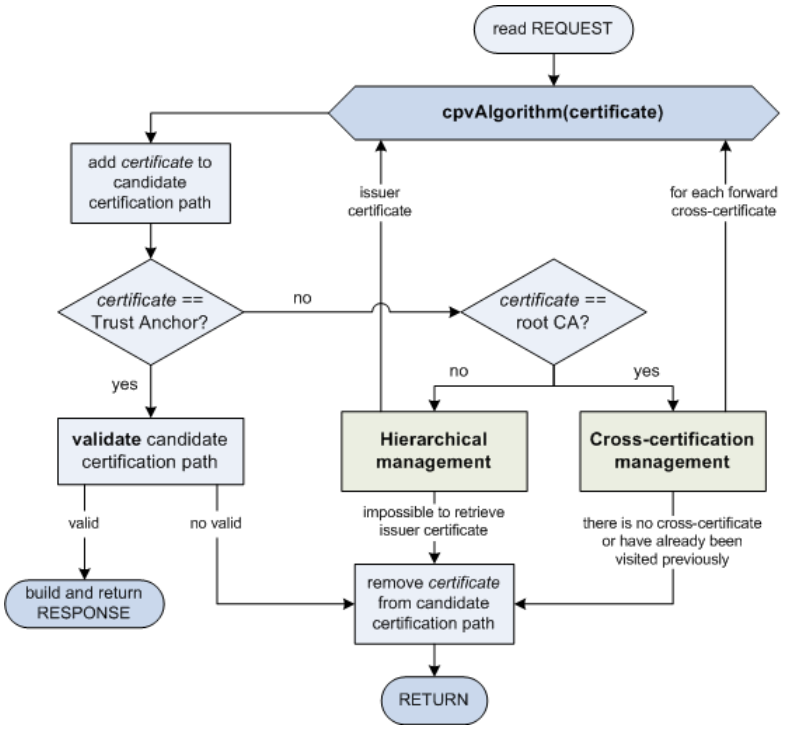

Figure 1: Main blocks of the certification path building and validation algorithm

This process can be interrupted when a full certification path is discovered. Thus, this candidate path should be validated. This validation supposes verifying both the content

of each certificate in the path (digital signatures, extensions, etc.) and their revocation status, which can be based in turn on two methods: CRL/ARL, gathered during the certification path building, or OCSP queries, where the algorithm has to perform an OCSP request/response operation for each certificate in the path.

Implementation and performance details of what kind of method should be deployed by each organization is described in the next sections.

4. Deployment of an inter-domain PKI scenario

Figure 2 illustrates the inter-domain PKI scenario deployed as a testbed both to assess how cross-certification models behave in a real situation and to analyze the impact of certification models and path lengths in the performance of complex scenarios. This analysis is presented in section 5.

It is important to note that this testbed is related with a real scenario deployed at the European level, in which the main objective has been to analyze the impact of the different mechanisms for certification path building and validation over the system performance.

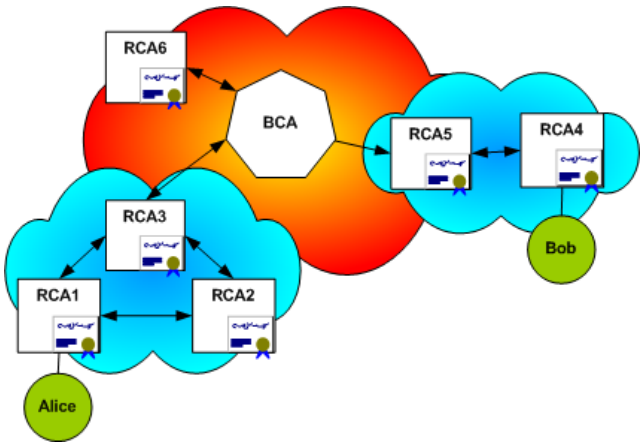

Figure 2: Inter-domain PKI testbed

This testbed is composed by six root CAs (RCA) and one BCA, using two different kinds of cross-certification models: Peer-to-Peer and Bridge CA. Let's suppose that Bob, belonging to RCA4, receives at any time some protected data, for example, a signed email message from Alice, who belongs to RCA1, and he would like to know if he can trust on it or not. That is, 1) there is a certification path between Alice's certificate and one on the Trust Anchors trusted by Bob, and 2) this certification path is valid regarding the content and revocation status. For that, Bob issues a protected SCVP request including the Alice's certificate and signs it with his own private key for being authenticated later by the server. This complete *SignedData* is sent to the VS of his domain (RCA4) in order to build and validate the best certification path. This algorithm has been described previously in section 3.3.

Depending on the Trust Anchors defined by Bob, the path length can vary from two certificates (Alice and RCA1, where RCA1 is the first Trust Anchor reached) to ten certificates for the largest path, supposing in this case that RCA4 is the only Trust Anchor for Bob. In this last case the certification path would be as follows:

{Alice}←{RCA1}←{RCA1↔RCA3}←{RCA3}←{RCA3↔BCA}←{BCA}←
{BCA→RCA5}←{RCA5}←{RCA5↔RCA4}←{RCA4}

Where {X} represents the public key certificate of end user or CA X, and {X↔Y} represents a cross-certificate agreed between entities X and Y.

4.1. PKI services and certificate extensions

To deploy this testbed, we have made use of the UMU-PKIv6 software [20, 21], which supports the definition of Hierarchical, Peer-to-Peer and Bridge CA cross-certification models. Every CA running the testbed has implemented, besides the LDAP repository, according to section 3.1, the following services:

- *OCSP and TSP servers*: these services have been implemented as Java Servlets by means of the Tomcat distribution (hardware and software details are described in the next section).
- *Validation Service*: this service has also been implemented as a Java Servlet and is offered by all CAs. It uses external information like CRLs and OCSP status information and supports the SCVP request/response protocol. The validation service, that implements the algorithm described in section 3.3, can be reached by end entities using a Java API that implements the SCVP protocols.

Regarding certificate extensions, we have followed the practical recommendations defined in section 3. In this case, the mandatory extensions have at least been included: *Authority*/*SubjectKeyIdentifier*, *KeyUsage* and *BasicConstraints*, which have been used to help the validation service to decide between different cross-certification paths, for example, when the next step to RCA1 has to be selected between RCA3 or RCA2, or between RCA5 and RCA6 for the next step to BCA; *AuthorityInfoAccess* has been also used to recover information about validation services (CRL/OCSP) from cross and end user certificates; and *NameConstraints* has been defined to exclude certification paths. Finally, policy extensions (*CertificatePolicies*, *PolicyMappings*, etc.) have also been defined, but its use is out of the scope of this work.

4.2. Software and hardware

Each CA depicted in Figure 2, either a RCA or the BCA, is set up on an independent server with the features indicated in Table 2.

For the validation process we have used the Java Certification Path API [22]. This API only supports the validation by means of CRLs, for what we have extended the *PKIXCertPathChecker* to also support the management of OCSP queries and responses.

5. Validation results

The scenario presented in section 4 has been deployed in a lab testbed with the aim of validating the trust management requirements described previously, and analyzing the performance impact of complex certification paths.

Table 2: Hardware and software requirements

Hardware	**CPU**	AMD Opteron 246 Microprocessor, 2.0 GHz, 32 bits
	Cache Size	1024 KB L2
	Total memory	1024 MB
	Hard disk space	>1 GB free
Software	**PKI**	UMU-PKIv6 7.2.1 Release Candidate 2
	Repository	OpenLDAP 2.3.27
	Database	PostgreSQL 8.1.14
	Servlet container	Apache Tomcat 5.5.17 (Servlet 2.4)
	JDK	Sun J2SE 1.5.0 Update 17, including the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 5.0
	Validation	Java Certification Path API

This section describes the kind of performance measurements that have been taken depending on these different factors:

- Revocation mechanism: CRL/ARL vs. OCSP.
- Certification path length: relying upon the defined Trust Anchor(s).
- Requests management: sequential vs. concurrent.

The main objective behind the tests is to asses how these factors affect in the building and validation of certification paths inside the inter-domain PKI testbed. All these tests contrast the performance of the VS depending on the revocation mechanism used, with the aim of showing how an off-line method behaves (CRL/ARL) against an on-line one (OCSP). Furthermore, we analyze the impact of changing the certification path length. Finally, the last test shows the behaviour of the VS when end users send their requests simultaneously. It will also provide us information about when this service overloads, if occurs.

5.1. Testing the Certification Path Building and Validation algorithm

Figure 3 depicts the five phases in which the algorithm described in section 3.3 has been divided, so we can assess the different time measurements. Each phase is labelled with a number, which will be used from now on to identify the specific process we are referring to.

- *Read request*: in this phase the VS receives the request signed by the user, including the certificate to be validated, the Trust Anchors in which the user trusts and what information the user wants from the service. Let's suppose for these tests that Bob requests (see Figure 2) the best certification path for Alice's certificate and the revocation outcomes.
- *Building logic*: recursive part of the algorithm which explores the certification tree finding the candidate certification paths according to the client requirements.
- *LDAP recovery*: this part is in charge of gathering from LDAP repositories the *pkiCA* object class element. This element contains all the necessary information

for finding the candidate certification paths, as well as the corresponding CRLs/ARLs for validation purposes.

- *Certification path validation*: during this phase, each certificate in the candidate path must be validated in two ways according to [4]. First, an off-line validation of the complete path such as certificates content integrity, critical extensions, signatures of the certification chain, etc. Second, the revocation status must be checked for each certificate in the path. This algorithm considers the next mechanisms:
  - CRL/ARL: these lists have already been retrieved in the previous phase.
  - OCSP: both OCSP requests and responses are digitally signed. Each OCSP server includes in its response a certifying time obtained from a timestamp token returned by a trusted TSP Authority. TSP requests and responses will also be signed to protect both messages.
- *Build response*: this response is digitally signed by the VS, including not only the reply status about the success of this process, but also the best certification path found and the corresponding validation outcomes.

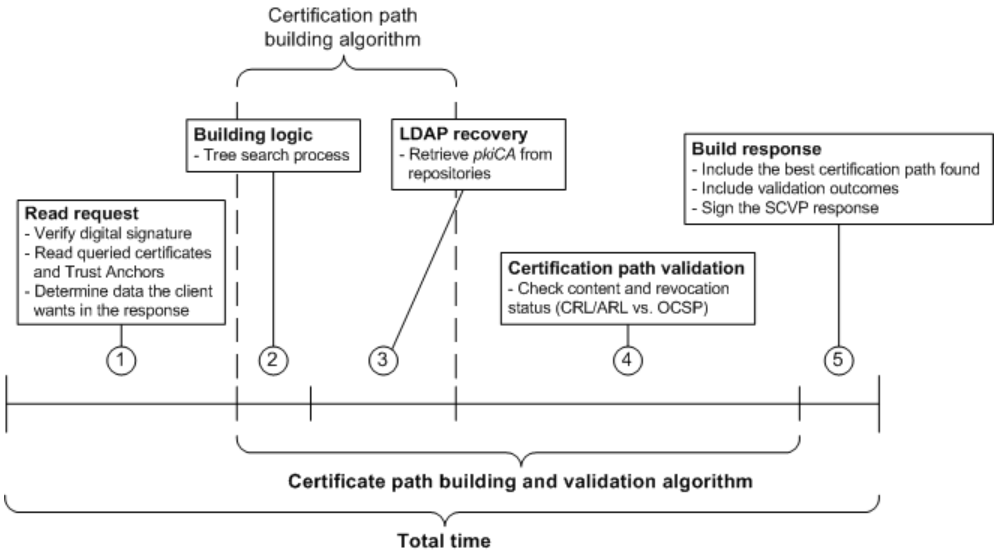

Figure 3: Partial processes of the Certification Path Building and Validation algorithm

Finally, it is worth noting that all measurements have been taken inside the RCA4's VS (Bob's domain), except the OCSP management. In this last case, times have been directly taken inside each OCSP Responder, which are called by the algorithm during the validation process.

5.1.1. Average time regarding the certification path length

This first test aims to assess the effect the length of the certification paths has on the different phases of the algorithm. It will give us an idea about the time consuming depending on the revocation mechanism (CRL/ARL or OCSP). Because of this, we analyze the behaviour of a Validation Service on cross-certification scenarios depending on the revocation method and the size of these certification paths.

The testing process has been carried out by means of sending 125 sequential requests, from which we have extracted the average time of each phase of the algorithm. These partial and total times are shown in Figure 4.

Figure 4a) shows how CRL/ARL mechanism affects on the performance of the VS, whereas Figure 4b) depicts the same information but using OCSP as revocation mechanism. In both column charts, the x-axis represents the Trust Anchor provided by Bob for the certification path building process, where RCA1 column is the shortest path (only Alice and RCA1 certificates), and RCA4 column is the largest one, 10 certificates from Alice to RCA4 (see section 4). The y-axis represents the times measured in milliseconds (ms). Also, each column shows the partial times taken by each phase of the algorithm. The number included on each column represents the total time, on average, the VS has taken for executing the algorithm.

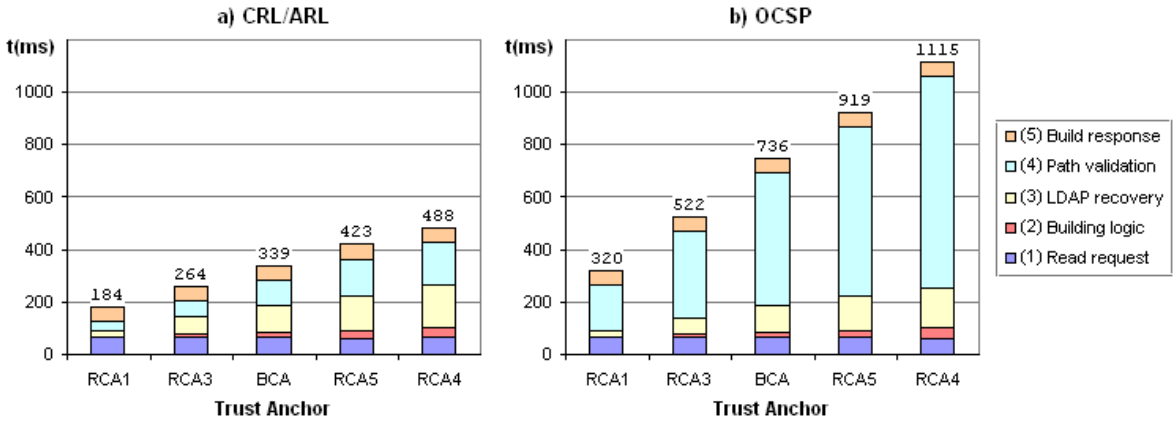

Figure 4: Processing average time depending on the certification path length

We can see that (1) and (5), which take 65 ms and 54 ms on average respectively, are values nearly constants and are independent of both the path length and the revocation mechanism. These phases could be considered out of the process since they are carried out just before and after executing the algorithm itself. On the other hand, we observe that phases (2), (3) and (4) can be considered linear times, i.e. O(n), with respect to the path length, being (4) the main factor during the validation of the candidate certification paths. Thus, and depending on the revocation mechanism, we can state that for CRL/ARL the increment of a new cross-certified CA in the inter-domain PKI testbed would suppose, on average, less than 100 ms more in the total processing time. This increment would suppose around 200 ms more for OCSP.

Analyzing both revocation methods, we could assert that CRL/ARL mechanism supposes an overload with respect the total time around 40%-55% lower than OCSP, although this percentage decreases as the certification path length grows, reaching a point where the validation process is stabilised around 45%. However, in a real environment this assumption can be slightly different (as we can see in section 5.2) since the OCSP requests depend in fact on the network traffic and delays, whereas in the CRL/ARL mechanism the time will be less important in percentage as all revocation information is harvested during phase (3) of the algorithm.

Finally, note that for the largest certification path (i.e. the worst case), the VS takes 488 ms for CRL/ARL mechanism and 1115 ms for OCSP. Both times are perfectly acceptable and can be assumed in any of the scenarios proposed in this paper.

5.1.2. Analysis of performance measurements for the largest certification path

Taking advantage of the previous tests, the goal of this section is to provide a more concise view in a numerical fashion about each of the different phases that constitute the algorithm. The times presented here correspond with the last column for both revocation mechanisms in Figure 4.

Table 3 collects the pertinent results for each phase of the algorithm, extending two of them in more atomic actions to be analyzed in detail. These two phases deal with the LDAP recovery and the revocation mechanism (both CRL/ARL and OCSP). Both the average time and the standard deviation are shown for each phase.

As can be seen, all times are similar independently of the revocation mechanism. The main difference between them lies on the validation process for managing OCSP queries, which increases the total time for this phase considerably: 166 ms for CRL/ARL against 807 ms for OCSP; in other words, OCSP takes almost five times more than CRL/ARL.

Table 3: Average time and standard deviation for the largest certification path

	(1)	(2)	(3)						(4)						(5)	TOTAL
			RCA1	RCA3	BCA	RCA5	RCA4	TOTAL	RCA1	RCA3	BCA	RCA5	RCA4	TOTAL		
Average time																
CRL/ARL	65	39	33	36	34	32	26	161	-	-	-	-	-	166	55	**488**
OCSP	63	37	34	35	36	33	25	163	159	146	146	149	127	807	54	**1115**
Standard deviation																
CRL/ARL	13	13	11	13	12	11	10	20	-	-	-	-	-	15	11	25
OCSP	11	13	10	11	12	10	9	19	25	25	22	24	18	65	9	70

As we have indicated at the beginning of this section, the OCSP management implies calling a TSP Authority in order to get a certifying time, which will be included in the OCSP response to indicate when it was produced. These calls take 37 ms on average; that is, around a quarter of the total time for the OCSP processing. Note that this particular value is not directly shown in Table 3 for simplicity.

Another observation we can see is the total time spent for the validation process using OCSP. We can observe that the sum of the partial times is 727 ms, and not 807 ms. This difference is the time the algorithm needs to check the content integrity of the certification path (mainly the digital signatures) and the time of sending the OCSP queries through the lab network.

5.1.3. Concurrent requests regarding the certification path length

This test aims to evaluate how the algorithm behaves when the VS receives a great amount of simultaneous requests. We take three factors to compare how the algorithm works and when it is overloaded: the revocation mechanism (CRL/ARL or OCSP); the certification path length; and the number of concurrent requests.

This test has been performed by sending a varying number of simultaneous requests, starting with 25 and adding 25 more in each test.

The times Figure 5 illustrates correspond with the real time clients wait for a response; that is, from the first request is received by the VS until the last one is sent to the corresponding client. The total times for the CRL/ARL mechanism are shown in Figure 5a), whereas the same information for OCSP queries is shown in Figure 5b). Each of those times, measured in milliseconds, represents the total processing time (y-axis) as the number of concurrent requests increases (x-axis). Each line illustrates the variation of the certification path length, which is done by changing the given Trust Anchor, as in the sequential testing presented before.

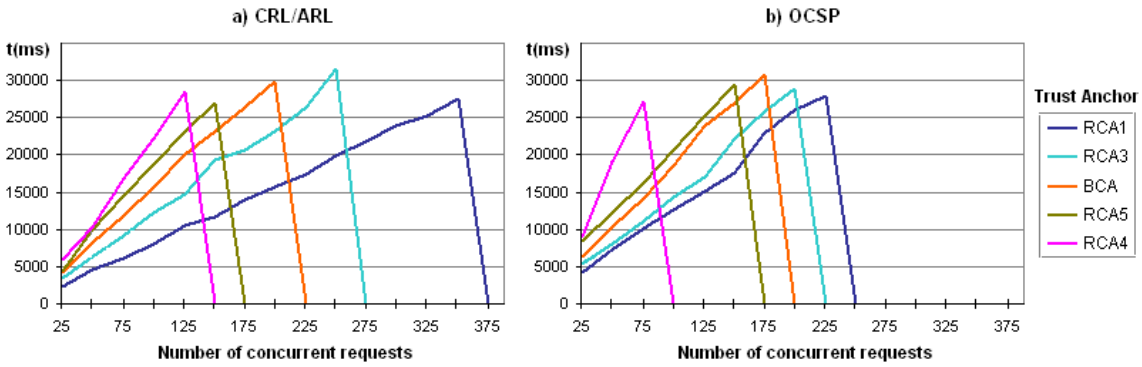

Figure 5: Processing real time depending on the concurrent number of requests and the certification path length

In this test we have increased the number of requests until reaching a threshold where an I/O exception is thrown, or even provoking a thread thrashing. This threshold has been set to 30 seconds. This means that, for the validation based on CRL/ARL, the VS crashes processing 375 requests simultaneously in the best case; that is, for the shortest certification path (Trust Anchor is RCA1). At worst, for the largest certification path (Trust Anchor is RC4), the VS crashes after receiving 150 requests. For OCSP mechanism, the outcomes are worst. In this case the VS can only handle until 250 concurrent requests for the shortest path, and 100 requests for the largest one.

Another observation we could consider is based on establishing a limit or threshold from which we deem users could abort the connection supposing a timeout. We have been considering this threshold of 10 seconds in this particular test. Bearing in mind this premise, and for the CRL/ARL mechanism, only 50 requests can be processed simultaneously before reaching that threshold when the Trust Anchor is either RCA4 or RCA5; 62 requests when the Trust Anchor is BCA; 83 requests for RCA3 and 118 requests for the shortest path (Trust Anchor is RCA1).

On the other hand, for OCSP validation, the expectation is worst taking into account that these tests have been carried out in a controlled lab environment. In this case, only 30 requests could be processed in the time indicated when the Trust Anchor is RCA4; 37 requests when Trust Anchor is RCA5; 49 requests for BCA; 65 requests for RCA3 and 74 simultaneous requests in the best of cases. Note that these particular values are not directly shown in Figure 5 for simplicity.

As a conclusion from these data, they can be assumed by any of the three main scenarios proposed in this paper. However, it would be advisable to do an in depth study depending on both the underlying inter-domain PKI scenario defined and which might be the largest certification path, since this last factor is the most important one that should be taken into account. These numbers represent bad results and we should find ways to speed up these processes, for example, caching some information such as

CRLs/ARLs, implanting workload balancing strategies for parallel processing through a computational grid, or even delegating some parts of the algorithm to others third trusted parties.

5.2. Time measurements in real environments

After analyzing times described before, which has been taken from a lab scenario, and observing how our VS behaves according to different proposed factors, it is unavoidable to ask how these times would be in a real scenario where network delays have a considerable influence. As commented in the previous section, only LDAP retrievals and OCSP operations depend on the network traffic and delays. The rest of the algorithm phases are executed locally inside the VS, thus being these phases independent of the scenario (lab or real). Thus, this section aims to asses how our VS would work taking different times from a real scenario by using public LDAP repositories and OCSP Responders from others important infrastructures.

After searching in more than 30 infrastructures, as the ones presented along section 2, Table 4 illustrates the average times and standard deviation for those infrastructures that provide public and free access to their LDAP and OCSP servers. Most of the infrastructures do not provide public access to their servers and, in some cases, only provide access to one of them (e.g. EuroPKI [23] offers a public OCSP Responder but not an LDAP repository). It is worth mentioning that this search has been mainly carried out through the home pages of each infrastructure, by looking for both inside the Certification Practice Statements (CPS), which should provide this information, and the extensions of each CA certificate.

Table 4: Average times and standard deviation in a real scenario

Domains	CROSS-CERTS		LDAP		OCSP	
	Forward	Reverse	Average time	Standard deviation	Average time	Standard deviation
DoD Root CA (1)	15	1	1446	95	184	14
DoD Interoperability Root CA (1)	3	1	1186	87	190	7
ORC Government ROOT (2)	4	1	821	74	280	26
ORC ROOT (2)	3	2	1024	92	268	21
SAFE-Biopharma Association	8	8	1519	79	309	35
Digital Signature Trust (DST)	1	1	762	67	446	58
EuroPKI	0	0	-	-	168	9
TOTAL (on average)			**1126**	**82**	**264**	**24**

(1) Root CAs belonging to the U.S. Defense Information Systems Agency's DoD PKI [24]
(2) Root CAs belonging to the ORC U.S. Government ECA [25]

This testing process has been carried by sending 25 sequential requests, both LDAP and OCSP, and extracting the average time and the corresponding standard deviation in milliseconds. Note that all these CAs, but EuroPKI, belong or are cross-certified with the Federal Bridge Certificate Authority (FBCA) [12].

As can be seen in Table 4, LDAP retrievals (1126 ms on average) suppose the most critical factor that any VS should take into account. For example, if we merge these times with the ones extracted in section 5.1.2 (for the largest certification path), the

complete process of building and validating that certification path would take almost 6 seconds in a real scenario, using CRL/ARL as revocation mechanism. That is, the five LDAP retrievals would take 5630 ms (1126 ms per retrieval), plus 325 ms that the VS would internally take to execute the rest of phases; this last value corresponds to the sum of the five phases less the one that represents the LDAP retrievals (3), which is replaced with this new real time. For OCSP, the complete process would take a bit more than 7 seconds; that is, 5630 ms in the five LDAP retrievals, and 1320 ms for the OCSP queries/responses (264 ms per each one) plus 234 ms for executing the rest of the phases. As before, this last time corresponds to the sum of all phases less times for the LDAP retrievals and the OCSP management, which have been replaced for the new real values. The time of checking the content integrity and sending the OCSP queries, as commented at the end of section 5.1.2, have also been added to this last time as part of the algorithm.

Anyway, the OCSP mechanism is not totally comparable to our OCSP times since these ones depend too much on which security features the OCSP server is providing. In our case, the OCSP servers require that all user requests are digitally signed for being authenticated, and their responses include a certifying time by a TSP Authority. These two features are not provided by any of the servers analyzed in this section.

Finally, just to observe that the column *cross-certs* has been included in Table 4 for analyzing whether the number of cross-certificates has some influence during the building process. It can be seen that as the number of cross-certificates increases the final time for LDAP retrievals also grows. This is due to the LDAP response length, which increases with this factor. For example, for Digital Signature Trust [26] domain, which takes 762 ms, the LDAP response takes up 5430 bytes. On the other hand, for SAFE-Biopharma Association [27] domain, which takes 1519 ms, the LDAP response takes up 27126 bytes.

6. Discussion

6.1. Building of a PKI federation

This section focuses on the underlying problematic about how current PKIs can join to existing ones, supposing that each of them has its own internal policies and might not accomplish with all the requirements, due to those internal policies, that another domain may consider as mandatory. This entails a lack of interoperability between these autonomous domains, doing that the new federation may have certain technical limitations. Due to this fact, every autonomous domain willing to form part of a global PKI federation must follow a set of certificate and service requirements, like those presented in section 3.2.

As an example, let’s suppose the Federal Bridge Certificate Authority (FBCA) as case study about interoperability between autonomous domains. The central BCA of this federation maintains 21 cross-certification relationships with the most important federal agencies. All the cross-certificates in this federation are well-defined according to the requirements presented in this work, at least they contain the *AuthorityInfoAccess* extension for building and validation purposes, but the *forward* cross-certificate issued by one of the root CAs joined to this federation. This means that relying parties will never be able to use this CA, or any of its subordinate certificates in its internal

hierarchy, as Trust Anchor during a validation process. That is, during the certification path building phase the VS reaches well the BCA by following the algorithm presented in section 3.3. Between all possible *forward* cross-certificates, there comes a point where the algorithm chooses the mentioned *forward* cross-certificate in order to reach the root CA. As such a certificate does not define the extension *AuthorityInfoAccess*, which indicates how and where to access to the information of that root CA, it will not be possible to build the complete certification path, thus providing a building error back to the user.

Thus, these federations must ensure compatibility and interoperability of their participant entities by following any kind of guide, recommendations or requirements like the ones presented in this work. In Table 1 we recommended the *AuthorityInfoAccess* extension as mandatory for every CA cross-certificate; others publications like [13] also strongly encourage supporting this extension in order to provide usability and interoperability with many existing PKIs. Even, the General Services Administration (GSA) federal agency, that performs proof of concept demonstrations for the FBCA, published a technical guidance and requirements matrices [28] to identify and resolve compatibility and interoperability problems that must be taken into account when new CAs wants to cross-certify with the FBCA. In this guide, the GSA promotes the inclusion of the extensions *AuthorityInfoAccess*, *SubjectInfoAccess* and *CRLDistributionPoints* with the proper encoded URIs to support path building and validation functionalities.

Apart from that, we have found in this federation another disagreement with our recommendations described in section 3.2. The majority of root CAs belonging to the FBCA infrastructure does not define the *SubjectInfoAccess* extension. This means that users cannot send to the VS the certificate of this root CA as the target one since the algorithm would be incapable of retrieving, for example, all its cross-certificates in order to follow the search until the defined Trust Anchor. This extension has been defined for us as mandatory, and it is also strongly advisable for both [13] and [28].

6.2. Certification path building and validation

Performance results derived from tests described in section 5, show that the total processing time depends above all on the certification path validation phase. Thus, the path length is the most important factor to take into account and becomes a critical component inside any certification infrastructure. For sequential processing, as commented in section 5.1.1, the inclusion of a new cross-certified CA to the certification path supposes to add two certificates more (CA certificate and the corresponding cross-certificate), thereby increasing the total processing time between around 100 ms (for CRL/ARL) and 200 ms (for OCSP), on average. In this case, the building and validation times are acceptable for large infrastructures. However, for concurrent processing, section 5.1.3, the building and validation times become unmanageable even for a medium organization. Some optimizations would have to be implemented to reduce that average time.

For medium-large organizations it is important to offer validation services with redundancy and load balancing techniques, in order to do not overload the service. Here network administrators have to take into account network design and databases distribution to ensure data synchronization between servers, as the authors recommend in [29] for traditional revocation mechanisms.

Other proposals can be found in [13], but they are mainly focused on graph theory to choose the best certification path when there are several alternatives; for example, when the VS retrieves several cross-certificates and the algorithm must decide the branch from which to continue to reach the defined Trust Anchor. Another classical optimization, also proposed in [13], is to use a local cache where storing locally at the VS all the needed information (e.g. certificates, CRLs/ARLs and/or OCSP responses) to speed up this process for future requests. This presents a serious drawback, since the VS could return invalid certification paths, if the previous information is not updated.

A proposal to solve this problem could be to adopt a hybrid solution during the validation process. The VS could follow a temporal validation approach during the first steps of the algorithm (first retrieved nodes) and deactivate this validation later. This approach relies on the assumption that the certification paths are usually invalid because the leaf nodes (end entities certificates) are not valid; they are normally revoked due to the private key has been compromised or lost, affiliation changes, etc. On the contrary, as the VS rises in the federation hierarchy it is more difficult to find invalid certificates. Thus, this partial validation could be carried out during the analysis both of end entities certificates (as commented above) and cross-certificates. The latter ones could be susceptible of modification when changing certain inter-domain policies, a new level of agreement is then agreed between two or more domains, and thus the extensions of these cross-certificates must reflect this change.

Another conclusion is that there are many infrastructures unable to provide OCSP services. For example, only 6 of 21 root CAs belonging to the FBCA can provide this kind of service. This supposes a stern drawback to infrastructures that want to provide online services to their clients. For example, organizations that rely on these infrastructures could be potentially exposed to fraud, theft and compromise, such as the ones we can find in e-government or e-commerce scenarios. In this case, these organizations will rely only on revocation status checking methods in an online way (e.g. OCSP responses) instead of using off-line revocation mechanisms (e.g. CRLs/ARLs or delta CRLs).

Because of this, all certificates in the certification path must be checked against the OCSP responder defined for each of them, thus providing a certain *level of trust* from which the mentioned organizations can offer more secure electronic transactions. As a conclusion, every entity belonging to a federation willing to offer more secure validation services will have to include in its certificates the *AuthorityInfoAccess* extension (access method *id-ad-ocsp*) to indicate the location of the OCSP responder. For example, as commented before, the FBCA cannot provide this kind of service since less than half of the infrastructures that constitute this federation provide OCSP responders.

7. Related work

There is not a common agreement on what kind of path building direction is better or worst. For example, [2] establishes that a mixed (forward and reverse) path building and validation algorithm would provide best solutions for complex certification infrastructures providing Hierarchical, Peer-to-Peer and Bridge CA relationships. However, few PKI solutions provide this kind of combined validation algorithm and most of them offer solutions based on forward or reverse directions, mainly due to the

complexity of programming and managing this kind of algorithm. This work makes use of the forward direction, as suggested by [30].

Lots of works have analyzed the performance of revocation mechanisms in PKI infrastructures [31,32], but mainly focused on timing for specific mechanisms, like CRLs, delta CRLs, or OCSP rather than the definition of complex certification scenarios and designing of building and validation algorithms.

In [30] the authors propose a framework able to model PKI protocols and services in network environments. Like most of the frameworks, simulators and testbeds, this framework is based on Java technology (SSFNet). Through this technology, the authors model PKI entities, protocols, data and networks, and provide a PathBuilder module which involves a certificate topology module, the algorithm for searching paths and defines criteria to distinguish between alternative branches. The main differences of this proposal are: first, they provide a simulation framework, where they predefine the certification topology and the services located at each organization. However, the proposal described in this paper presents the performance of a real certification building and validation algorithm implementation; second, they provide performance measurements based only on CRL mechanisms and run only sequential validation request, but not concurrent ones. The authors propose a certification topology composed by five BCAs, trying to simulate the establishment of trust connections between FBCA, EuroPKI, and other institutions. The testbed simulates 51 PKI domains and 103 ordinary CAs. On average, for a path length of 3.6 on the forward direction they obtain a building delay of 7.7 seconds, taking into account the network latency. According to the results derived from section 5.2, for the forward algorithm presented in section 3.3, and a path length value of 5, the average time is 6 seconds, which slightly improves the results of [30].

8. Conclusions and future work

Public Key Infrastructures have become one of the main components to deploy trust and security models inside those organizations that want to protect communication channels or access to resources and internal data, among others. This situation, together with the proliferation of scenarios like identity federations, e-business, etc., where organizations need to establish trust relationships among them, provides new requirements. In these scenarios, PKIs need to be flexible enough to be adapted to these new requirements, which may include in some cases the establishment of cross-certification relationships among CAs which are usually the source of trust of each organization.

This paper presents the design, implementation and validation of an inter-domain PKI infrastructure that could be instantiated to any of these scenarios. In this infrastructure a set of non-hierarchical relationships between security domains is established to obtain a trustworthy model. Once these trust relationships have been created, certificates need to be defined with certain mandatory, required and/or optional extensions. Some clear procedures also need to be defined for building and validating certification paths.

As a statement of direction we are currently working on the provision of similar trust models for roaming environments, where one user can move between several security domains, and network and service providers need to validate his/her certificate before allowing him/her to access to the network and/or starting a secure communication with him/her.

Acknowledgements

This work has been partially funded by the MISTRAL (Middleware de gestión de Identidades de Seguridad en TRansacciones electrónicAs basado en código Libre, TIC-INF 07/01-0003) project. This work has been also partially funded by the CENIT SEGUR@ (Seguridad y Confianza en la Sociedad de la Información) project. Authors would also like to thank the Funding Program for Research Groups of Excellence with code 04552/GERM/06 granted by the Fundación Séneca.

References

1. S. Kiran, P. Lareau, S. Lloyd. “PKI Basics - A Technical Perspective”, PKI Forum, White Paper, November 2002.
2. S. Lloyd, Ed. “CA-CA Interoperability”, PKI Forum, White Paper, March 2001.
3. P.M. Hesse, D.P. Lemire. “Managing Interoperability in Non-Hierarchical Public Key Infrastructures”, Proceedings of the Network and Distributed System Security Symposium (NDSS), February 2002.
4. D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, IETF RFC 5280, May 2008.
5. Euro6IX EU-IST Project Home Page, http://www.euro6ix.org
6. The InCommon federation, http://www.incommonfederation.org
7. The SWITCH federation, http://www.switch.ch/aai
8. K. Wierenga et al. “Inter-NREN Roaming Architecture. Description and Development Items”, GN2 JRA5 (GEANT 2), September 2006. Project Deliverable DJ5.1.4.
9. Internet2 Salsa-FWNA (Federated Wireless NetAuth) Sub-Group, http://security.internet2.edu/fwna
10. European Bridge-CA (EB-CA), http://www.bridge-ca.org
11. Chinese Taipei BCA, http://www.nii.org.tw
12. Federal Bridge Certification Authority (FBCA), http://www.cio.gov/fbca
13. M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, R. Nicholas. “Internet X.509 Public Key Infrastructure: Certification Path Building”, IETF RFC 4158, September 2005.
14. D. Pinkas, R. Housley. “Delegated Path Validation and Delegated Path Discovery Protocol Requirements”, IETF RFC 3379, September 2002.
15. T. Freeman, R. Housley, A. Malpani, D. Cooper, W. Polk. “Server-Based Certificate Validation Protocol (SCVP)”, IETF RFC 5055, December 2007.
16. M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams. “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP”, IETF RFC 2560, June 1999.
17. Department of Information Security and Electronic Signature, Slovakian National Security Authority. “Certificate Path Validation v1.4”, No. 1891/2006/IBEP-011, November 2006.
18. G. Martínez Pérez, F. García Clemente, M. Gil Pérez, A.F. Gómez Skarmeta. “Secure Overlay Networks for Federated Service Provision and Management”, Computers & Electrical Engineering, 34(3):173-191, May 2008.
19. Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman, S. Proctor. “Building Certification Paths: Forward vs. Reverse”, Proceedings of the Network and Distributed System Security Symposium (NDSS), February 2001.
20. A.F. Gómez Skarmeta, G. Martínez Pérez, O. Cánovas Reverte, G. López Millán. “PKI Services for IPv6”, IEEE Internet Computing, 7(3):36-42, June 2003.

21. UMU-PKIv6 Home Page, http://pki.inf.um.es
22. S. Mullan. “Java Certification Path API Programmer’s Guide”, February 2003. http://java.sun.com/j2se/1.5.0/docs/guide/security/certpath/CertPathProgGuide.html
23. EuroPKI Home Page, http://www.europki.org
24. DoD PKI, Defense Information Systems Agency, U.S. Department of Defense, http://iase.disa.mil/pki
25. U.S. Government External Certificate Authority (ECA), http://www.eca.orc.com
26. Digital Signature Trust (DST), IdenTrust Inc. http://www.identrust.com
27. SAFE-Biopharma Association, http://www.safe-biopharma.org
28. General Services Administration. “FBCA and C4CA Cross-Certification”, Technical Guide, March 2007.
29. A.J. Slagell, R. Bonilla, W. Yurcik. “A Survey of PKI Components and Scalability Issues”, Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC), pp. 475-484, April 2006.
30. M. Zhao. “Performance Evaluation of Distributed Security Protocols Using Discrete Event Simulation”, Dartmouth Computer Science Technical Report TR2005-559, PhD Thesis, October 2005.
31. A. Aarnes, M. Just, S.J. Knapskog, S. Lloyd, H. Meijer. “Selecting Revocation Solutions for PKI”, Proceedings of the Fifth Nordic Workshop on Secure IT Systems (NORDSEC), October 2000.
32. J. Iliadis, S. Gritzalis, D. Spinellis, D. De Cock, B. Preneel, D. Gritzalis. “Towards a Framework for Evaluating Certificate Status Information Mechanisms”, Computer Communications, 26(16):1839-1850, October 2003.