Building a Virtual Hierarchy for Managing Trust Relationships in a Hybrid Architecture

Cristina Satizábal
Telematics Engineering Department, Technical University of Catalonia, Barcelona, Spain
Email: isabelcs@entel.upc.edu
Engineering and Architecture Department, Pamplona University, Pamplona, Colombia

Rafael Páez, Jordi Forné
Telematics Engineering Department, Technical University of Catalonia, Barcelona, Spain
Email: {rpaez, jforne}@entel.upc.edu

***Abstract*—Trust models provide a framework to create and manage trust relationships among the different entities of a Public Key Infrastructure (PKI). These trust relationships are verified through the certification path validation process, which involves: path discovery, signature verification and revocation status checking. When trust relationships are bidirectional, multiple paths can exist between two entities, which increase the runtime of the path discovery process. In addition, validation of long paths can be difficult, especially when storage and processing capacities of the verifier are limited. In this paper, we propose a protocol to establish a hierarchical trust model from a hybrid PKI. This protocol makes more efficient certification path discovery since in a hierarchical model, trust relationships are unidirectional and paths are easy to find. In addition, our protocol does not require issuing new certificates and allows setting a maximum path length, so it can be adapted to the features of users' terminals.**

***Index Terms*—certification path discovery, hierarchical trust model, hybrid trust model, Public Key Infrastructure (PKI).**

I. Introduction

One of the most important aspects of any business transaction is trust. In Internet, where there is not direct contact between the parties and millions of users interoperate, identity theft is frequent and it is necessary to adopt security measures that allow us to authenticate our business partners, consumers and suppliers, previous to the interchange of information, goods and services.

Public Key Infrastructure (PKI)[1] provides the required trust using Trusted Third Parties (TTPs) known as Certification Authorities (CAs). These digitally sign data structures called Public Key Certificates (PKCs), ensuring that a particular public key belongs to a certain user. Thus, certificates and their keys give the connecting parties information about their business partners.

However, the mere existence of a certificate does not guarantee its authenticity. The recipient of a certificate must verify its signature and validity before trusting the certificate's content. If the same CA issued the certificates of the communicating parties, one can easily verify the signature of the other's certificate using the public key of this CA. However, to verify the signature of a certificate issued by another CA, it is necessary certain trust relationship among the PKI authorities. There are different ways to establish such trust relationship called trust models. These allow a user to build chains of certificates from its trusted CA to the other users known as certification paths.

When trust relationships are unidirectional, such as a hierarchical architecture, paths are well defined and are easy to find. However, if trust relationships are bidirectional, such as mesh architecture, all possible options do not lead to the target entity and there can be multiple paths between two entities. This makes difficult the task of the verifier and increase the runtime of certification path discovery.

Validation of long paths can be difficult too, as much from the computational point of view, since public key algorithms require complex mathematics calculations, like considering the set of resources necessary to obtain, store and verify the certificates. Thus, verifiers with limited capacities, such as mobile devices, may not be able to offer enough resources to carry out this process.

The purpose of our proposal is to take advantage of the efficiency in the path discovery process offered by hierarchical trust model. For that reason, our protocol establishes a virtual hierarchy among the authorities of a hybrid PKI. In addition, the hierarchy is built considering a maximum path length, whose value can be established taking into account the features of the users' terminals.

Sections II and III describe certification path validation process and PKI trust models, respectively. In section IV, we present some existing solutions to increase the efficiency of certification path discovery process in decentralized architectures. The operation of our protocol is described in section V. Section VI shows a practical example of our protocol. In section VII, we compare certification path length of a hybrid PKI and the hierarchy

obtained with our protocol. Finally, section VIII concludes.

This paper is an extended version of [2]. Here, we give a more detailed description of our protocol and introduce a practical example to clarify its operation.

II. Certification Path Validation

A CA's certification domain defines the organizational or geographical boundaries within which the CA is considered trustworthy. Thus, all the PKI users in a CA's certification domain consider this authority like their trust anchor.

A trust anchor is a certification authority that a PKI user explicitly trusts under all circumstances; this is used by the client application as the starting point for all certificate validation. Each user receives the public key of its trust anchor when it is registered in the PKI.

When two users belong to the same certification domain and they want to communicate each other, one can obtain easily the other's public key, since they know the public key of their trust anchor. But when users belong to different certification domains their communication is only possible if there is an uninterrupted chain of trust points between them, which supposes the intervention of several CAs and an agreement among their policies. CAs use cross certification to allow users building trust chains from one point to another known as certification paths.

Cross certification is the establishment of a trust relationship between two certification authorities through a certificate signed by a CA that contains the public key of another CA, referred to as cross certificate [1].

A certification path [3] is a chain of public key certificates through which a user can obtain the public key of another user. The path is traced from the verifier's trust anchor to the CA public key required to validate the target entity's certificate. Thus, the certification path length is equal to the number of CAs in the path plus one: a certificate for each CA and the target entity's certificate.

The primary goal of a path validation process is to verify the binding between a subject and a public key. Then, the verifier must check the signature of each certificate in the path in order to trust the public key of the target entity. In general, a path validation process involves the following steps:

- *Discovering a Certification Path:* It is to build a trusted path between the verifier's trust anchor and the target entity based on the trust relationship among the CAs of the PKI. When a certification path is built from the target entity to a trust anchor, this is called building in the forward direction. When a certification path is built from a trust anchor to the target entity, this is called building in the reverse direction [4].
- *Retrieving the Certificates:* It is to retrieve each certificate in the path from the place(s) where they are stored. The most common method for the distribution of certificates and certificate revocation information in the enterprise domain is publication. The idea behind publication is that PKI information is posted in a widely known, publicly available, and easily accessible location. Publication is particularly attractive for large communities of users who in general are personally unknown to one another (that is, the PKI information does not have to be distributed directly to each individual). In today's enterprise, it is common practice to post (or publish) certificates and certificate revocation information (particularly revocation information based on CRLs [1]) to a repository. A repository is a generic term used to denote any logically centralized database capable of storing information and disseminating that information when requested to do so[5].
- *Verifying the Digital Signatures:* It is to verify the validity of the digital signature of each certificate in the path. It involves:
  1. Decrypting the signed part of the certificate with its issuer's public key.
  2. Calculating a hash of the certificate's content.
  3. Comparing the results of 1 and 2. If they are the same then the signature is valid.
- *Verifying the Validity of the Certificates:* It is to determine if the certificates have expired or have been revoked. The certificate validity period is used to verify the expiration, while the revocation status depends on the revocation mechanism used. Certificate revocation is the mechanism under which an issuer can revoke the binding of an identity with a public-key before the expiration of the corresponding certificate. Issuer can use periodic publication mechanisms such as Certificate Revocation Lists (CRLs)[1], or on-line query mechanisms such as the Online Certificate Status Protocol (OCSP)[6].

III. Trust Models

Certification architectures or trust models provide a technological framework for creating and managing trust relationships among the different entities of a PKI. They describe how trust relationships and the necessary rules to find and to cross certification paths are built. The most common PKI trust models are: a single CA, hierarchical, mesh, bridge CA and hybrid [7], [8], [9].

A. Single CA Model

In this model, all the PKI users trust the only CA of the architecture (Fig. 1). Therefore, each certification path begins with the CA's public key.

This configuration is the simplest to deploy, but if the CA's public key changes, all the architecture must be reconfigured. In addition, this model is not suitable for very large or diverse communities of users.

B. Hierarchical Model

It is the most common model. In this configuration, all the users trust the same root CA (RCA). Thus, a user of a hierarchical PKI begins its certification paths with the RCA's public key. In a hierarchical PKI, trust relationships are unidirectional, that is, subordinate CAs do not issue certificates to their superior CAs (Fig. 2).

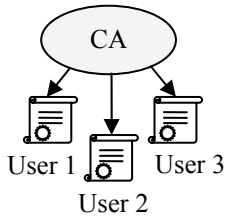

Figure 1. Single CA model

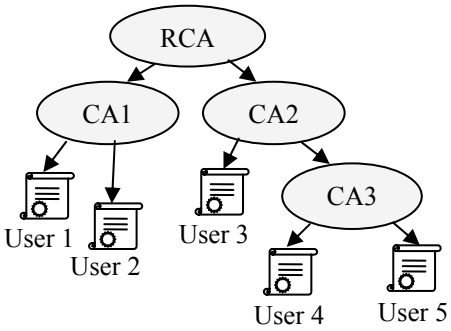

Figure 2. Hierarchical model

Hierarchical PKIs are scalable. Certification paths are easy to find because they are unidirectional and the longest path is equal to the depth of the tree less one, since RCA certificate is not part of the path. In addition, users of a hierarchy know implicitly which applications a certificate may be used for, based on the position of the CA within the hierarchy.

Hierarchical model has a single trust point, so if the RCA's private key is known by another entity, all the PKI is put at risk. In addition, transition from a set of isolated CAs to a hierarchical PKI may be logistically impractical because all users must adjust their trust points.

C. Mesh Model

It is also known as cross-certificate architecture. Here, the trust anchor of a user is its local CA and all the CAs are autonomous, so a CA does not rely on a superior CA. An autonomous CA can perform peer-to-peer cross-certification with other autonomous CAs. Thus, a pair of certificates describes their bidirectional trust relationship (Fig. 3). However, trust relationships may not be unconditional. If a CA wants to limit its trust, the authority must specify these limitations in the certificates it issues. All certificate validation, by clients of an autonomous CA, starts with the local CA certificate.

Mesh PKI can easily incorporate a new community of users and although management cost is high, there is not a single point of failure and multiple paths can exist between two users. In addition, a mesh PKI can easily be constructed from a set of isolated CAs because users do not need to change their trust points.

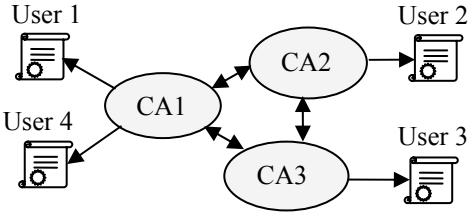

Figure 3. Mesh model

In this model, the number of trust relationships is directly proportional to the number of CAs, what brings scalability problems. In addition, users must determine which applications a certificate may be used for based on the content of the certificates, so certificates have more extensions and path validation process is more complex.

The maximum path length in a mesh model is the number of CAs in the PKI.

D. Bridge CA Model

William Polk and Nelson E. Hastings [8] use a Bridge Certification Authority (BCA) to establish a peer-to-peer trust relationship among different user communities, acting like a hub (Fig. 4).

Users know their path to the BCA and they need only to determine the path from the BCA to the target entity. In this model, certification path discovery is easier than in a mesh PKI although it is more complex than in a hierarchical PKI. The BCA must use certificate information to establish trust relationships among different enterprise PKIs. Thus, certificates are more complex and PKI users must be prepared to process and use this additional information during validation of certification paths. A technical challenge of BCA based PKIs, which has largely been ignored, is the distribution of certificates and certificate status information in a way useful to users and their applications. In an effective PKI, users must be able to readily obtain CA and user certificates and the corresponding certificate status information from a distribution mechanism.

E. Hybrid PKI

The dynamic nature of the business relationships, Internet, etc., make desirable that trust models are not static and closely limited. As its name indicates, hybrid model allows mixing the previous certification architectures, for example, it is possible to connect a hierarchical PKI with a single CA model through cross-certification between their CAs (Fig. 5)

Thus, relationships among user communities will determine which model is appropriate.

IV. Related Works

There are different proposals which increase efficiency of path discovery and certification path validation in decentralized architectures using a hierarchical model.

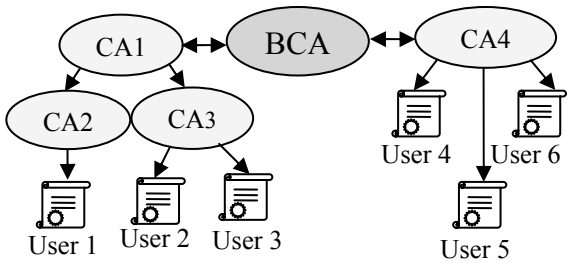

Figure 4. Bridge CA model

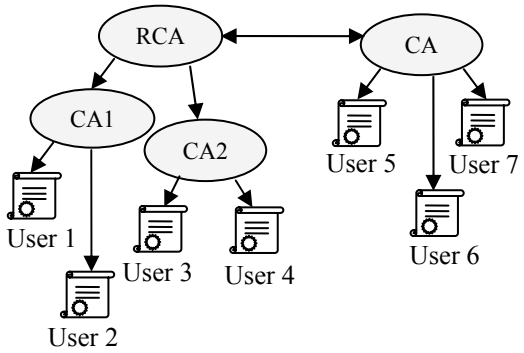

Figure 5. Hybrid model

Marchesini and Smith [10] propose a virtual hierarchy formed in a peer-to-peer network, that allows verifying certification paths in an efficient manner. The established trust chains are loop-free, and the secrets (private keys) are splitted in multiple fragments, so that the compromise of one of them does not affect all the architecture obtaining resilient trust chains. Thus, resulting nodes are virtual CAs, formed by several authorities that share a portion of a private key. Each node represents the collective action of a set of conventional CAs.

Pan et al. [11] propose a scheme of merging multiple PKIs, which is based on a hierarchical structure too. This scheme includes two phases. In the first phase, a special certificate authority is selected as the new root certification authority from the set of CAs. In the second phase, all the other certification authorities must apply to the root CA for their new certificates. These new certificates still use the previous public keys. Thus, the merging process is quick and low-cost, and the certificate path processing is much more simple and efficient than using cross-certification. However, the selection of the root CA is not clearly defined by authors because this is an administrative action, which needs to be decided concerning the reality of the new enterprise.

Unlike these proposals, our protocol uses the existing trust relationships among PKI authorities to create the virtual hierarchy, so it is not necessary to issue new certificates, adjust the trust points or create new nodes during the protocol execution. In addition, since validation of long paths is difficult for verifiers with limited processing and storage capacity, our protocol sets a maximum certification path length whose value can be established taking into account the features of the users' terminals.

V. Protocol Description

In this section, we describe the operation of our protocol. This establishes a virtual hierarchy among the authorities of a hybrid PKI, based on the trustworthiness level of the participant entities.

Thanks to unidirectional trust relationships of hierarchical models, our protocol improves the efficiency of the certification path discovery process in applications that require digital signature verifications.

In addition, since validation of long paths is difficult for verifiers with limited capacities, the hierarchy is built considering a maximum path length, whose value can be set taking into account the features of users' terminals.

In our protocol, hierarchy is built from the leaves to the root (upwards). Theoretically, the most trustworthy authority of the architecture will be the root CA, but actually, it is not always possible to establish what authority will be the root CA of the hierarchy, because each authority only receives information from its neighbors, so the hierarchy can have more than one root CA. Therefore, if the hierarchy were built from the root to the leaves (downwards), they would be necessary a greater number of messages among the authorities of the PKI to determine at the beginning of the protocol which authority is the root CA. For that reason, we build the hierarchy upwards. Thus, the less trustworthy authorities choose first a superior CA among their trust CAs and the most trustworthy authorities become superior CAs of the hierarchy.

Some aspects of our protocol are inspired on the algorithm proposed by J. Hernandez-Serrano et al in [12], although the application area is different. Table I shows the notation used in this paper.

We divide our protocol in two phases to understand it better.

- *Trustworthiness order among CAs:* In this phase, neighboring CAs are arranged from the less trustworthy to the most trustworthy, based on two parameters that characterize them.
- *Construction of the hierarchy:* In this phase, it is established a hierarchical trust relationship among CAs of the hybrid PKI.

A. Trustworthiness Order among CAs

The protocol begins when an authority CA_0 of a hybrid PKI declares to its neighbors (authorities that issued a certificate to CA_0 and authorities that CA_0 issued a certificate) that it wants to establish a hierarchical trust relationship with them. In addition, CA_0 propose a maximum certification path length (L_{MAX}) based on the processing and storage capacity of its users.

Thus, CA_0 sends a request message to its neighbors containing the value of L_{MAX}. These messages and all the messages sent among PKI authorities along the protocol must be authenticated by the receiver.

Each neighbor can accept or refuse to collaborate in the establishment of that hierarchy, sending to the demanding authority an acceptance or rejection message.

TABLE I.
NOTATION

Notation	Meaning
L_{MAX}	Maximum path length allowed
CA_i	Certification authority i
L_i	Number of certificates from the leaves to the authority i
IN_i	Number of authorities which CA_i trusts (received certificates)
OUT_i	Number of authorities that trust CA_i (issued certificates)
CA_0	Current authority
N_0	Number of participant neighbors of CA_0
Order[N_0+1]	Array that contains CA_0 and its participant neighbors ordered from the less trustworthy to the most trustworthy
pos	Position of CA_0 inside the Order array

Once authority CA_0 receives the responses from all its neighbors, it determines the number of authorities that want to be part of the hierarchy and issued a certificate to CA0 (IN_0), and the number of authorities that want to participate in the hierarchy and received a certificate from CA0 (OUT_0). Later, CA_0 sends these values to its participant neighbors in an information message and these neighbors send to CA_0 their own parameters IN_i and OUT_i.

Later, CA_0 compares OUT_0 with the received OUT_i values and puts them in order from the lowest to the highest. The authority with the lowest OUT_i is the less trustworthy, that is, the neighbor that less the other participants trust. If there are two or more authorities with the same OUT_i, they are arranged in accordance with the IN_i value from the lowest to the highest too. For the sake of simplicity, we have not considered other parameters to put in order the authorities such as existing policy mapping or distance among them, but these can be considered if parameters OUT_i and IN_i are the same for two or more authorities. Thus, each authority put in order its neighbors, from the less trustworthy to the most trustworthy, determining which of its neighbors are less trustworthy and more trustworthy than itself. At the beginning of the protocol, L_i=0 for all the authorities.

The following algorithm describes the procedure done by each authority in the first phase of the protocol.

```
BEGIN
    IF CA0 begins the protocol THEN
      CA0 CHOOSE L_MAX
      CA0 SEND request message TO all its neighbors
      CA0 RECEIVE acceptance/rejection messages
      FROM its neighbors
    ELSE
      CA0 RECEIVE request message FROM some
      neighbor
      IF CA0 does not want to be part of the hierarchy
      THEN
        CA0 SEND rejection message TO its
        demanding neighbor
        CA0 finishes the protocol
        GO TO END
      ELSE
        CA0 SEND acceptance message TO its
        demanding neighbor
        CA0 SEND request message TO its other
        neighbors
        CA0 RECEIVE acceptance/rejection messages
        FROM its neighbors
      END IF
    END IF
    CA0 COMPUTE IN0 and OUT0
    CA0 SEND IN0, OUT0 TO its participant neighbors
    CA0 RECEIVE INi, OUTi FROM its participant
    neighbors
    Order[1]=0 /*CA0 in position 1 of Order array*/
    pos=1
    FOR j=1 TO N0
      IF OUT0 < OUTj THEN
        GO TO POST
      ELSE IF OUT0 = OUTj THEN
        IF IN0 <= INj THEN
POST:     CA0 is less trustworthy than CAj
          FOR p=N0+1 TO pos+2
            Order[p]=Order[p-1]
          END FOR
          Order[pos+1] = j
          FOR k=(pos+1) TO N0
            p1=Order[k]
            p2=Order[k+1]
            IF OUTp1 > OUTp2 THEN
              p1 is more trustworthy than p2
              Order[k]=p2
              Order[k+1]=p1
            ELSE IF OUTp1 = OUTp2 THEN
              IF INp1 > INp2 THEN
                p1 is more trustworthy than p2
                Order[k]=p2
                Order[k+1]=p1
              END IF
            END IF
          END FOR
        ELSE
          GO TO PREV
        END IF
      ELSE IF OUT0 > OUTj THEN
PREV:   CA0 is more trustworthy than CAj
        pos = pos+1
        FOR p=N0+1 TO pos
          Order[p]=Order[p-1]
        END FOR
        Order[pos-1] = j
        IF (pos-1)>1
          FOR k=(pos-1) TO 2
            p1=Order[k]
            p2=Order[k-1]
            IF OUTp1 < OUTp2 THEN
              p1 is less trustworthy than p2
              Order[k]=p2
              Order[k-1]=p1
            ELSE IF OUTp1 = OUTp2 THEN
              IF INp1 <= INp2 THEN
                p1 is less trustworthy than p2
```

```
                    Order[k]=p2
                    Order[k-1]=p1
                  END IF
                END IF
              END FOR
            END IF
          END IF
        END FOR
END
```

B. Construction of the Hierarchy

In this phase of the protocol, authorities act from the less trustworthy to the most trustworthy in accordance with the order established at the first phase. Therefore, the less trustworthy authority in the neighborhood acts first and the other authorities must wait for the intervention of their less trustworthy neighbors.

The objective of the second phase is that each authority chooses a superior CA among the participant neighbors that issued it a certificate (trusted neighbors). Thus, when an authority CA_0 acts, it looks for the most trustworthy authority of its trusted neighbors, based on the trustworthiness order established at the first phase of the protocol, and chooses this neighbor like superior CA. If L_0 is higher than L_i of superior CA and (L_0 +1) is less than or equal to (L_{MAX} − 1), L_i of superior CA takes the value of (L_0 +1). In case that (L_0 +1) is higher than (L_{MAX}−1), the chosen superior CA is not appropriate and CA_0 must choose the next trusted neighbor like superior CA provided that this neighbor is more trustworthy than CA_0. CA_0 checks again if L_0 is higher than L_i of the new superior CA and so on until CA_0 finds a suitable superior CA. Nevertheless, it can be possible that none of the trusted neighbors that are more trustworthy than CA_0 can be used like superior CA. Thus, when CA_0 concludes this procedure, it sends an association message to its neighbors informing the identity of its superior CA or a failure message if it was not possible to choose a superior CA.

Later, the following less trustworthy authority in the neighborhood, according to the order established in the first phase, repeats the procedure and so on until all authorities act, except for the most trustworthy authority of the neighborhood that must not carry out this procedure because there is not a neighbor more trustworthy than it.

The authorities that did not choose a superior CA in this phase of the protocol, including the most trustworthy authority, are considered root CAs. If there are more than one root CA at the end of the second phase, the protocol must be repeated with the resulting root CAs, considering only the certificates issued among them to determine the new value of OUT_i and IN_i parameters. Thus, a root_CA message is the response to a failure message. The root_CA message must be sent only by the root CAs at the moment that they act. L_i maintain the value that they obtained during protocol execution. In addition, when the protocol is repeated, the value of L_i can be less than or equal to L_{MAX} in the second phase, instead of (L_{MAX} − 1).

Even so, hierarchy can have more than one root CA after the repetition of the protocol. In this case, the root CAs must find the shortest path among them using an alternative method.

Root CAs send their public key to all the authorities below them once the protocol has concluded (root_CERT message).

The following algorithm describes the procedure done by each authority in the second phase of the protocol.

```
BEGIN
    IF protocol repetition THEN
      L_M = L_MAX
    ELSE
      L_M=L_MAX-1
    END IF
    IF CA_0 is the most trustworthy neighbor THEN
      CA_0 is root CA
      IF CA_0 RECEIVE failure message THEN
        CA_0 RETURN root_CA message
        root CAs REPEAT the protocol
      ELSE
        CA_0 is the root of the hierarchy
        CA_0 SEND root_CERT message TO all PKI
        authorities
      END IF
    ELSE IF CA_0 is the less trustworthy neighbor THEN
      GO TO ACT
    ELSE
      CA_0 RECEIVE association message FROM its
      less trustworthy neighbors
ACT:  FOR j = N_0+1 TO pos+1
        t=Order[j]
        IF CA_0 trust CA_t THEN
          n = L_t
          IF L_t <= L_0 THEN
            L_t=L_0+1
          END IF
          IF L_t <= L_M THEN
            CA_0 CHOOSE CA_t like superior CA
            BREAK /*exit FOR*/
          ELSE
            L_t = n
            CA_0 must choose another superior CA
          END IF
        END IF
      END FOR
      IF CA_0 did not choose a superior CA THEN
        CA_0 SEND failure message TO its participant
        neighbors
        CA_0 is root CA
      ELSE
        CA_0 SEND association message TO all its
        neighbors
      END IF
    END IF
END
```

VI. PRACTICAL EXAMPLE

Fig. 6 shows a hybrid PKI with 10 authorities, randomly generated. Arrows represent certificates issued from one authority to another. Table II also represents this hybrid PKI, where '1' means that there is a certificate between two entities.

Authority 1 wants to carry out our protocol, so it sends a request message to its neighbors (2, 4, 5, 6, 8 and 10) and proposes a maximum path length L_{MAX}=2.

If authority 2 wants to collaborate with authority 1, it sends an acceptance message to 1 and a request message containing the value of L_{MAX} to its other neighbors (3, 5, 7, 8, 9 and 10).

For the sake of simplicity, we suppose that all nodes want to be part of the hierarchy. Thus, at the same time, authority 4 sends an acceptance message to authority 1 and a request message to authorities 3, 5, 8, 9 and 10; authority 5 sends an acceptance message to authority 1 and a request message to authorities 2, 3, 4, 7, 9 and 10; authority 6 sends an acceptance message to authority 1 and a request message to authorities 8 and 10; authority 8 sends an acceptance message to authority 1 and a request message to authorities 2, 3, 4, 6, 7, 9 and 10; and authority 10 sends an acceptance message to authority 1 and a request message to authorities 2, 3, 4, 5, 6, 7 and 8.

Later, authority 3 sends an acceptance message to authorities 2, 4, 5, 8 and 10, and a request message to authority 9. At the same time, authority 7 sends an acceptance message to authorities 2, 5, 8 and 10, and a request message to authority 9; authority 9 sends an acceptance message to authorities 2, 4, 5 and 8, and a request message to authorities 3 and 7; authority 2 sends an acceptance message to authorities 5, 8 and 10; authority 4 sends an acceptance message to authorities 5, 8 and 10; authority 5 sends an acceptance message to authorities 2, 4 and 10; authority 6 sends an acceptance message to authorities 8 and 10; authority 8 sends an acceptance message to authorities 2, 4, 6 and 10; and authority 10 sends an acceptance message to authorities 2, 4, 5, 6, and 8.

Finally, authority 3 sends an acceptance message to authority 9. At the same time, authority 7 sends an acceptance message to authority 9; and authority 9 sends an acceptance message to authorities 3 and 7.

When authorities receive response to their request messages, they must determine their OUT_i and IN_i values and send them to their neighbors. Table III shows the parameters of each node.

Once each authority obtains the parameters of its neighbors, puts them in order from the less trustworthy to the most trustworthy. Thus, for authority 1 the trustworthiness order is: 6, 2, 4, **1**, 8, 10, 5.

Likewise, the other authorities determine their trustworthiness order. For authority 2 is: 7, 3, **2**, 1, 8, 9, 10, 5; for authority 3 is: **3**, 2, 4, 8, 9, 10, 5; for authority 4 is: 3, **4**, 1, 8, 9, 10, 5; for authority 5 is: 7, 3, 2, 4, 1, 9, 10, **5**; for authority 6 is: **6**, 1, 8, 10; for authority 7 is: **7**, 2, 8, 9, 10, 5; for authority 8 is: 7, 3, 6, 2, 4, 1, **8**, 9, 10; for authority 9 is: 7, 3, 2, 4, 8, **9**, 5; for authority 10 is: 7, 3, 6, 2, 4, 1, 8, **10**, 5.

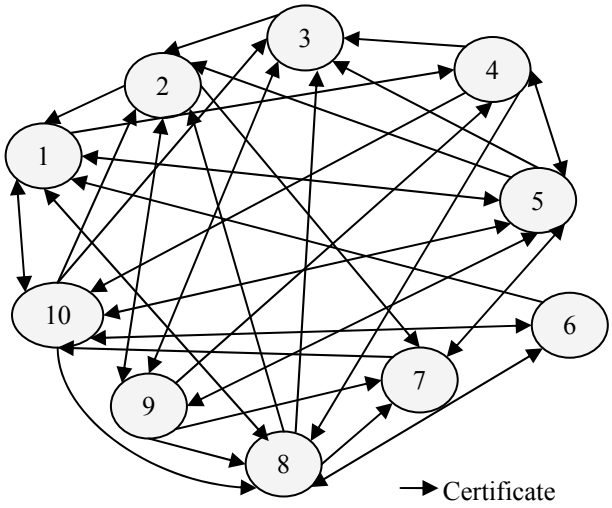

Figure 6. Hybrid PKI

TABLE II.
CONNECTIONS AMONG THE CAs

FROM \ TO	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	1	0	0	1	0	1
2	1	0	0	0	0	0	1	0	1	0
3	0	1	0	0	0	0	0	0	1	0
4	0	0	1	0	1	0	0	1	0	1
5	1	1	1	1	0	0	1	0	1	1
6	1	0	0	0	0	0	0	1	0	1
7	0	0	0	0	1	0	0	0	0	1
8	1	1	1	0	0	1	1	0	0	0
9	0	1	1	1	1	0	1	1	0	0
10	1	1	1	0	1	1	0	1	0	0

TABLE III.
PARAMETERS OF THE CAs

	1	2	3	4	5	6	7	8	9	10
OUT_i	4	3	2	4	7	3	2	5	6	6
IN_i	5	5	5	3	5	2	4	5	3	5

According to this order, authorities 3, 6 and 7 act first, then authorities 2 and 4, next authority 1, later authority 8, after that, authorities 9 and 10. Finally, authority 5 is the most trustworthy.

Authority 6, among its trusted neighbors (8 and 10), chooses 10 like superior CA, since it has the highest OUTi. Since, $L_{10}=L_6=0$ and $(L_6+1)=(L_{MAX}-1)$, $L_{10}=L_6+1=1$. Therefore, authority 6 sends an association message to its neighbors, indicating that authority 10 is its superior CA.

The most trustworthy neighbor for authorities 3 and 7 is 5, so they choose this authority like superior CA. Then, $L_5=1$.

Authority 1 trusts authorities 2, 5, 6, 8 and 10, but authority 5 is the most trustworthy among them. Thus, authority 1 chooses 5 like superior CA.

Now, authority 8 acts. It chooses authority 10 like superior CA.

Then, among its trusted neighbors (2, 3 and 5), authority 9 chooses authority 5.

At the same time, authority 10 chooses its superior CA. Among its trusted neighbors (1, 4, 5, 6 and 7), authority 5 is the most trustworthy, but $L_5=L_{10}=L_{MAX}-1=1$, so

authority 5 is not a suitable superior CA for 10. Thus, authority 10 sends a failure message to its neighbors and authority 5 returns a root_CA message. Therefore, authorities 5 and 10 are root CAs of the hierarchy and they must repeat the protocol.

Since, authority 5 trusts 10 and authority 10 trusts 5, when the protocol is repeated OUT_5=OUT_{10}=1 and IN_5=IN_{10}=1. In this case, it is necessary to consider another parameter to determine which authority is the most trustworthy. Since more authorities chose authority 5 like superior CA at the first iteration of the protocol, we consider that authority 5 is more trustworthy than 10. Therefore, authority 10 chooses authority 5 like superior CA and L_5=2.

Fig 7 shows the established hierarchy. Authority 5 is the root CA, so it must send a root_CERT message to its subordinated authorities.

VII Certification Path Length Comparison

In this section we compare the length of certification paths in the hybrid PKI of previous section (Fig. 6) and the hierarchy obtained with our protocol (Fig. 7).

Path length is the number of certificates that a verifier must check to authenticate another entity. Table IV shows the length of the shortest path between two authorities in the hybrid PKI. A path length equal to '0' means that the current CA knows the public key of the other authority, so it is not necessary to verify any certificate.

Table V shows the length of certification paths in the established hierarchy. The paths that increase their length compared with Table IV are enhanced. The length of the other paths is less than or equal to the path length in Table IV.

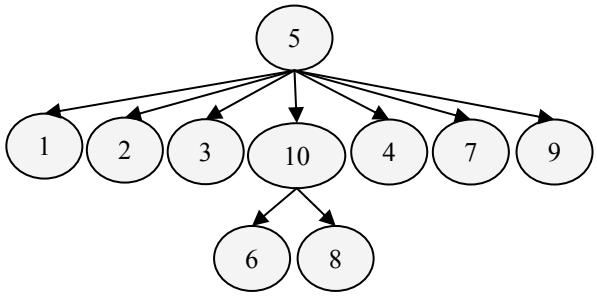

Figure 7. Established hierarchy among CAs

TABLE IV.
LENGTH OF THE SHORTEST CERTIFICATION PATHS

FROM \ TO	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	1	0	1	0
2	1	0	0	1	0	1	1	0	0	0
3	1	1	0	0	0	1	1	0	0	0
4	0	1	1	0	0	2	1	1	0	1
5	0	1	1	0	0	1	0	1	0	0
6	1	1	1	2	1	0	1	0	2	0
7	1	0	1	1	0	1	0	0	0	1
8	0	1	1	0	1	0	1	0	0	0
9	1	0	0	1	0	2	1	1	0	1
10	0	1	1	0	0	0	0	1	1	0

TABLE V.
LENGTH OF CERTIFICATION PATHS IN THE HIERARCHY

FROM \ TO	1	2	3	4	5	6	7	8	9	10
1	0	**1**	1	1	0	**2**	1	**2**	1	**1**
2	1	0	**1**	1	0	**2**	1	**2**	**1**	**1**
3	1	1	0	**1**	0	**2**	1	**2**	**1**	**1**
4	**1**	1	1	0	0	2	1	**2**	**1**	1
5	**1**	1	1	**1**	0	**2**	**1**	**2**	**1**	**1**
6	1	1	1	1	0	0	1	**1**	1	0
7	1	**1**	1	1	0	**2**	0	**2**	**1**	1
8	**1**	1	1	**1**	0	**1**	1	0	**1**	0
9	1	**1**	**1**	1	0	2	1	**2**	0	1
10	**1**	1	1	**1**	0	**1**	**1**	1	1	0

VIII. Conclusions

Hybrid architecture is adaptable to the dynamic nature of Internet because it mixes unidirectional and bidirectional trust relationships. However, bidirectional trust relationships increase the complexity of the path discovery process, since there can be multiple paths between two entities and all the paths do not lead to the target entity. In addition, validation of long paths can be difficult, as much from the computational point of view, like considering the set of resources necessary to obtain, store and verify the certificates. Thus, verifiers with limited capacities, such as mobile devices, may not be able to offer enough resources to carry out this process.

In this paper, we propose a protocol that establishes a virtual hierarchy in a hybrid PKI, based on the trustworthiness of the participant CAs. The trustworthiness level of each CA is determined in accordance with two parameters: the number of certificates that it issues to the participant CAs (OUT_i) and the number of certificates that it receives from the participant CAs (IN_i).

An advantage of this protocol is that it does not establish new trust relationships among the CAs but it takes the existing certificates to establish the hierarchy. Thus, it is not necessary to issue new certificates among the authorities.

The establishment of a hierarchy among the CAs facilitates the path discovery process, since there is only one path between two entities. Therefore, paths are easy to find in the forward direction (from the target entity to the trust anchor). Thus, the efficiency of certification path validation is increased.

Also, our protocol is adaptable to users with limited processing and storage capacities, since hierarchy is established considering a maximum certification path length (L_{MAX}) that can be set according to the features of the users' terminals.

Our protocol does not always find a single root CA, what not implies that there is not a path among the authorities. For that reason, in those cases, we advise to use alternative methods to find the shortest path among the resulting root CAs.

Comparison of Tables IV and V shows that in the hierarchy obtained with our protocol, the certification path length between two different authorities diminishes or remains equal in most of the cases. However, the

average path length increases since the certificates that are not part of the hierarchy are omitted.

Future work will consist of introducing improvements to the protocol so that authorities can choose the shortest path to the root and the hierarchies have only one root CA, whenever there is a path among the authorities. Also, we will evaluate the efficiency of our protocol, compared with other proposed solutions. In addition, we will look for an appropriate way to guarantee that the exchanged information is truthful. Also, we must give some kind of incentive to the authorities, so that they want to participate in the hierarchy.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Research Council under the projects ARPA (TIC2003-08184-C02-02) and SECONNET (TSI2005-07293-C02-01)

REFERENCES

[1] ITU-T, "Recommendation X.509: Information Processing Systems - Open Systems Interconnection - the Directory : Authentication Framework (Technical Corrigendum)", International Telecommunication Union, 2000.

[2] C. Satizábal, R. Páez and J. Forné, "PKI Trust Relationships: from a Hybrid Architecture to a Hierarchical Model", *The First International Conference on Availability, Reliability and Security (ARES 2006),* IEEE Computer Society, Vienna University of Technology (Austria), 2006, pp. 563-570.

[3] R. Housley, W. Polk, W. Ford and D. Solo, "RFC3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", 2002.

[4] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman and S. Proctor, "Building Certification Paths: Forward vs. Reverse", *Network and Distributed System Security Symposium (NDSS 2001)*, 2001.

[5] C. Adams and S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*, Addison-Wesley, 2003.

[6] M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, "RFC2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", 1999.

[7] J. Linn, "Trust Models and Management in Public-Key Infrastructures", RSA Laboratories, 2000, http://www.rsasecurity.com/rsalabs/.

[8] W. T. Polk and N. E. Hastings, "Bridge Certification Authorities: Connecting B2B Public Key Infrastructures", NIST, 2000.

[9] R. Perlman, "An Overview of PKI Trust Models", *IEEE Network*, 1999, vol. 13, pp. 38-43.

[10] J. Marchesini and S. Smith, "Virtual Hierarchies - An Architecture for Building and Maintaining Efficient and Resilient Trust Chains", *7th Nordic Workshop on Secure IT Systems (NORDSEC 2002)*, Karlstad (Sweden), 2002.

[11] H. Pan, J. Li, Y. Zhu and D. Wei, "A Practical Scheme of Merging Multiple Public Key Infrastructure in E-commerce", *Networking and Mobile Computing: 3rd International Conference (ICCNMC 2005),* Springer-Verlag, Zhangjiajie(China), 2005, pp. 1287-1294.

[12] J. Hernandez-Serrano, J. Pegueroles and M. Soriano, "GKM over Large MANET", *IEEE International Workshop on Self Assembling Wireless Networks (SAWN2005)*, 2005, pp. 484-490.

Cristina Satizábal was born in Popayán (Colombia) on July 5th, 1976. She received her degree in electronic and telecommunications engineering from Cauca University (Colombia) in 2000. Currently, she is carrying out a PhD. in telematics engineering at the Technical University of Catalonia (Spain).

She is a teacher at the department of engineering and architecture of Pamplona University (Colombia). Her research interest includes Public Key Infrastructure (PKI), Privilege Management Infrastructure (PMI) and Intrusion Detection Systems (IDS). She has written around 10 papers for national and international conferences and journals.

Rafael Páez was born in Colombia on August 7th ,1972. He received his degree as systems engineer from the Catholic University (Colombia) in 2001. Also, he carried out graduate studies in security of data processing networks at the Catholic University (2002).

He has experience as network administrator and security tools, firewalls and Intrusion Detection Systems. On the other hand, he has been the author and speaker of national and international papers during his PhD.

Mr. Páez has interest about the security of data processing networks, especially Intrusion Detection Systems (IDS), Public Key Infrastructure (PKI), Privilege Management Infrastructure (PMI) and perimeter security.

Jordi Forné was born in Barcelona (Spain) in 1967. He received the MS. and PhD. in telecommunications engineering from the Technical University of Catalonia (Spain) in 1992 and 1997, respectively.

Currently, he is an Associate Professor at the Telecommunications Engineering School, Technical University of Catalonia, Barcelona (Spain). He has been working in cryptography and information security for the last 15 years. His research interests include network and multimedia security, electronic commerce and Telematics services. He has authored over 60 papers in international journals and conferences.