An Efficient, Dynamic and Trust Preserving Public Key Infrastructure

Albert Levi
Bogazici University, Department of Computer Engineering, Bebek, 80815 Istanbul, Turkey
Oregon State University, Electrical and Computer Engineering Department, 97331 Corvallis, Oregon, USA
levi@ece.orst.edu *levi@boun.edu.tr*

M. Ufuk Caglayan
Bogazici University, Department of Computer Engineering, Bebek, 80815 Istanbul, Turkey
caglayan@boun.edu.tr

Abstract

Nested certification is a methodology for efficient certificate path verification. Nested certificates can be used together with classical certificates in the Public Key Infrastructures (PKIs). Such a PKI, which is called Nested certificate based PKI (NPKI), is proposed in this paper as alternative to classical PKI. The NPKI formation model is a transition from an existing PKI by issuing nested certificates. Thus, we can extract efficiently verifiable nested certificate paths instead of classical certificate paths. NPKI is a dynamic system and involves several authorities in order to add a new user to the system. This uses the authorities' idle time to the benefit of the verifiers. In this paper, we analyze the trade-off between the nested certification overhead and the time improvement on the certificate path verification. This trade-off is acceptable in order to generate quickly verifiable certificate paths. Moreover, PKI-to-NPKI transition preserves the existing hierarchy and trust relationships in the PKI, so that it can be used for strictly hierarchical PKIs.

1. Introduction

Public Key Infrastructure (*PKI*) is the indispensable part of a public key cryptography-based application. A PKI is a certificate network used to find the correct public key for users. A *certificate* is a digitally signed binding between the public key and the real identity of a user. Certificates are issued by trusted *Certificate Authorities* (*CAs*). The verifier verifies the digital signature of the CA over the certificate and finds the correct public key for the certified user. However, in a PKI, there are several CAs and the verifier cannot know the public key of each CA. Therefore, the verifier spends most of its time to verify a *certificate path* with several certificates derived from PKI in order to find the public key of a user. The last certificate on the path is the certificate of the user whose public key is being sought. Each certificate on a path is verified to find the public key of the next CA and each public key is used to verify the next certificate. The verifier has to know the public key of the first CA and has to trust all CAs on the path.

Several PKIs are proposed in the literature. Most of them are based on the ISO/ITU-T X.509 [1] certificate standard. Privacy Enhanced Mail (PEM) [8] is the first functional X.509 based system. It is intended to create confidential and authentic e-mail transfer between its users. Secure Electronic Transaction (SET) [9] is yet another X.509 based system. The United States Postal Service (USPS) initiated the Information-Based Indicia Program (IBIP) [10] to support new public key cryptographic methods for using postal services on the Internet. The proposed infrastructure for IBIP is a three level X.509 based hierarchical PKI. Public Key Infrastructure for X.509 certificates (PKIX) [11], [12] is a general certificate infrastructure. Secure/Multipurpose Internet Mail Extensions (S/MIME) [13] is a secure Internet mail system. The certification infrastructure of S/MIME is based on PKIX infrastructure. Chokhani [14] proposed an X.509 based national PKI. There are also non-X.509 based PKIs. Pretty Good Privacy (PGP) [2] e-mail security system seems to have the most widely used PKI of this category. The Simple Public Key Infrastructure (SPKI) [15], [16], Simple Distributed Security Infrastructure (SDSI) [17] and Domain Name System SECurity extensions (DNSSEC) [18] are other examples of non-X.509 based PKIs.

Although the X.509 standard does not enforce any topology for a standard PKI, X.509 based PKIs are generally hierarchical and centralized. The general characteristics of an X.509 based PKI are (i) strict distinction between a CA and the end user (that is, the end users cannot issue certificates), (ii) a tree hierarchy with 3-7 levels and (iii) forming optional CA networks via cross certificates (not applicable for PEM). Moreover, the roles and responsibilities of the CAs in the specific levels are well defined in the PKI specifications and most of the time it is not possible to bypass a level in the hierarchy. PEM and SET are very strict on this issue. Although it is not X.509 based, DNSSEC also uses a hierarchical PKI.

Another important phenomenon for the PKI and certification systems is the "trust" concept. The verifier must trust the CA in order to verify a certificate issued by it. Although CAs are known as trusted entities, the verifiers must be able to choose their trusted CAs. In the X.509 based systems, every CA is a potentially trusted entity, but there are some mechanisms to avoid "blind trust" to the CAs. In the third version of X.509 [1], *policy identifiers* were added to the certificate structure as an optional extension for user initiated trust management. Trust manipulation is also taken into account in other PKI systems.

We call the certificate and PKI systems discussed above, X.509 based or not, "classical", as opposed to the "nested certification" system which is introduced below.

1.1. The focus of the paper

The classical certification systems use public key cryptography in order to digitally sign and verify the certificates. Public key cryptography operations are generally time inefficient. Moreover, all certificates on the path must be verified one by one in order to verify the certificate of a target user. The verifier only wants to find the correct public key of the target entity, but it has to verify the certificates of all intermediate CAs on the path and find their public keys in order to reach the target entity. We think that this is an unnecessary process degrading time efficiency.

One way of improving certificate path verification time is to let each CA verify the public keys of the end users via certificate paths and issue *direct classical certificates* for them. In this way, the verifiers who want to find the public key of an end user can quickly verify a direct classical certificate instead of following a certificate path. PGP [2] and ICE-TEL [7] use similar approaches. However, there are three shortcomings of this approach. The first shortcoming is that all CAs on the certificate path must be trusted in order to verify the path and to issue a direct certificate. If even one of the CAs is not trusted, the path cannot be verified and the direct certificate cannot be issued. The second shortcoming stems from the strict hierarchical structures of some PKIs. Direct certificate issuance spoils the hierarchical structures of these PKIs. Therefore, such PKIs do not allow direct certificate issuance. Finally, the third shortcoming is the increase in the number of certificates.

In this paper, we propose a PKI, which is named as *Nested certificate based PKI* (*NPKI*), from which it is possible to extract efficiently verifiable certificate paths. Although NPKI is based on a new *nested certification* [19] concept, both classical and nested certificates are used together. The proposed NPKI model is a *transition from an existing PKI*. Extra nested certificates are created during this process, but the initial topology of the source PKI is preserved in all stages of the transition process. Moreover, the transition to NPKI does not assume or require any trust relationships among the entities of the PKI. Therefore, it is possible to apply the NPKI transition even if there is no trust information in hand. These two characteristics of the NPKI model make it superior to the direct classical certification. Our primary focus is on the PKIs with a hierarchical backbone.

The nested certification concept and the nested certificate paths are overviewed in Section 2. In Section 3, the proposed transition from an existing PKI model is detailed. Performance evaluation of the proposed method is given in Section 4. Moreover, the nested certification overhead is analyzed and the trade-off between this overhead and efficiency improvement is interpreted in this section. Section 5 gives the conclusions and possible future work.

2. Nested certification

In this section, the nested certificate structure, subject certificate verification method and the nested certificate paths will be explained. This section is an extract from [19].

2.1. Definitions and terminology

In simple terms, a nested certificate is defined as "*a certificate for another certificate*". A classical certificate gives assurance about the correctness of the binding between the identity and the public key of an entity. Therefore, it is verified to find the correct public key of the certified entity. A nested certificate, on the other hand, certifies another certificate by assuring the legitimacy of the signature over it. Therefore, nested certificates are used to verify the signatures over other certificates. For example, certificate 1 is a classical certificate in Figure 1, since it is issued by *A* to verify the public key of *B*. Certificates 2 and 3 are nested certificates in Figure 1, since they are issued to certify other certificates. Certificate 2 is issued by *C* to certify certificate 1. Similarly, certificate 3 is issued by *D* to certify certificate 2. *Nested Certification Authority (NCA)* is the authorized issuer of a nested certificate. For example, *A* is a CA, *C* and *D* are NCAs in Figure 1. The certificate, which is certified by a nested certificate, is called *subject certificate*. For example, certificate 1 is the subject certificate of nested certificate 2, in Figure 1. Similarly, certificate 2 is the subject certificate of certificate 3.

Subject certificate is not a new certificate type. Any classical or nested certificate can be a subject certificate. For example, in Figure 1, the subject certificate 1 is a classical certificate and the subject certificate 2 is a nested certificate.

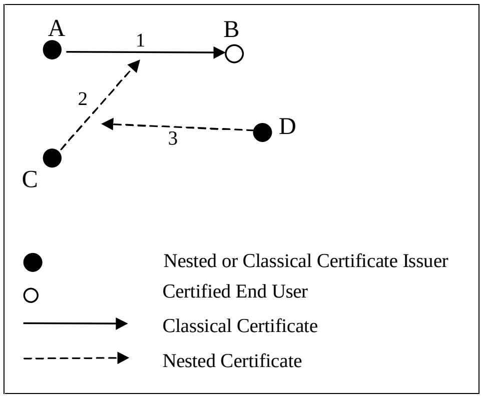

Figure 1. The certification relationships

2.2. Structure

An NCA issues a nested certificate by digitally signing the one way hash of the nested certificate content. The content of a nested certificate is related to its requirements. The two requirements of a nested certificate are:

(i) to certify that the subject certificate content has been signed by the claimed CA or NCA and

(ii) to certify that the subject certificate content has not been maliciously modified.

In order to satisfy the first requirement, the nested certificate contains the existing signature over the subject certificate content. In order to satisfy the second requirement, the nested certificate contains the hash of its subject certificate content. This hash can be obtained by applying an irreversible one way hash function [4], [5] to the subject certificate content. In order to issue a legitimate nested certificate, the NCA must have verified the signature over the subject certificate content successfully beforehand.

Formal representation of a nested certificate is given below. The notation used in this representation and the subsequent formal representations is given in Table 1. Given a subject certificate SC = $\mathrm{Cnt}_{SC}|\mathrm{Sig}_{SC}$, a nested certificate for SC issued by *NCA* is denoted as:

$$\mathrm{NC}_{NCA}(\mathrm{SC}) = \mathrm{Cnt}_{\mathrm{NC}}|\mathrm{Sig}_{\mathrm{NC}} = \mathrm{Cnt}_{\mathrm{NC}}|NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]],$$

where $\mathrm{Cnt}_{\mathrm{NC}} = schash|scsig|Other$, $schash = \mathrm{H}[\mathrm{Cnt}_{\mathrm{SC}}]$, which is the subject certificate hash, $scsig = \mathrm{Sig}_{\mathrm{SC}}$, which is the subject certificate signature and *Other* is the other managerial fields such as algorithms used, serial numbers, etc.

Table 1. The notation used in formal representations

Notation	**Meaning**
X_p	The public key of *X*
X_s	The secret (private) key of *X*
X_s[I]	The signature of *X* over the information I. The signature is issued by X_s.
X_p[I]	Inverse operation of X_s[I]. Expected to return I′, if I= X_s[I′].
H[I]	Hash (message digest) of information I.
$I_1\|I_2$	Concatenation of the information I_1 and I_2.
NC_{NCA}(SC)	The nested certificate, which is issued by *NCA*, for the subject certificate SC.
$\mathrm{Cnt}_{\mathrm{cert}}$	Content of certificate cert (does not include the digital signature).
$\mathrm{Sig}_{\mathrm{cert}}$	Signature over H[$\mathrm{Cnt}_{\mathrm{cert}}$].

It is extremely important to realize that the NCA does not guarantee the correctness of information of the subject certificate by issuing a nested certificate. The NCA guarantees the legitimacy of the signature over the subject certificate and conveys this information to the verifiers; no trust information is conveyed by a nested certificate. Therefore, in order to issue a nested certificate, the NCA need not trust anyone, even the subject certificate issuer. In this way, the nested certificates become an alternative to direct classical certificates and they can be issued where direct certification is not possible.

2.3. Cryptographic nested certificate verification method

In this method, the digital signature over the nested certificate content is verified by employing public key cryptosystem based signature verification operations. Figure 2 gives the cryptographic nested certificate verification algorithm.

```
Given:  A nested certificate,
        $\mathrm{NC}_{NCA}$ (SC) = $\mathrm{Cnt}_{\mathrm{NC}}|NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]]$ issued by a
        trusted NCA and the correct public key
        of NCA, $NCA_p$.

The verifier applies the following algorithm to verify NC.

Verified_Hash $\leftarrow NCA_p[NCA_s[\mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]]]$
Calculated_Hash $\leftarrow \mathrm{H}[\mathrm{Cnt}_{\mathrm{NC}}]$
IF   Calculated_Hash = Verified_Hash   THEN
   NC becomes verified
ELSE
   NC has not been verified
```

Figure 2. Cryptographic nested certificate verification algorithm

Verification of a nested certificate returns information about the correct hash value and signature over its subject certificate content. This information is used for the verification of the subject certificate using the *subject certificate verification* method.

2.4. Subject certificate verification method

The information returned by nested certificate verification is not sufficient to verify its subject certificate. By the verification of a nested certificate, only the correct hash value and correct signature over the subject certificate are found. In order to verify the subject certificate, the actual hash and the actual signature over the subject certificate must be compared with the ones stored in the nested certificate. Verification of a certificate as the subject certificate of a nested certificate is called *subject certificate verification*. Although the subject certificate verification method is a consequence of the nested certification, it can be used to verify both nested certificates and classical certificates, since the subject certificates can be of both types.

Given: a subject certificate, SC = $\mathrm{Cnt}_{SC}|\mathrm{Sig}_{SC}$, issued by a trusted authority and a legitimate nested certificate for SC,
$\mathrm{NC}_{NCA}(\mathrm{SC}) = \mathrm{Cnt}_{NC}|\mathrm{Sig}_{NC} = \mathrm{Cnt}_{NC}|NCA_s[\mathrm{H}[\mathrm{Cnt}_{NC}]]$,

where Cnt_{NC} contains the fields *schash*, which is equal to $\mathrm{H}[\mathrm{Cnt}_{SC}]$, and *scsig*, which is equal to Sig_{SC}.

The verifier applies the following algorithm to verify SC.

```
$Calculated\_Hash \leftarrow \mathrm{H}[\mathrm{Cnt}_{SC}]$
IF  $Calculated\_Hash = \mathrm{Cnt}_{NC}.schash$  AND
    $\mathrm{Sig}_{SC} = \mathrm{Cnt}_{NC}.scsig$  THEN
        SC becomes verified
ELSE
        SC has not been verified
```

Figure 3. Subject certificate verification algorithm

The subject certificate verification algorithm is given in Figure 3. As can be seen from this algorithm, having verified the nested certificate, *NC*, in order to verify its subject certificate, *SC*, the verifier follows the following two steps:

(a) The hash of the content of the actual *SC* is recalculated. This recalculated hash must be the same as the one stored within the *NC*.

(b) The actual signature over the content of the *SC* is compared with the subject certificate signature stored in the *NC*. These two signature values must be the same.

If the conditions given above are met and the verifier trusts the issuer of *SC*, then the verifier concludes that the *SC* is legitimate. The verifier must trust the issuer of *SC*, because the verification of *SC* does not mean that the information stored within *SC* is correct.

The subject certificate verification method does not use public key cryptography operations. Therefore, it is more efficient than public key cryptography-based certificate verification. Moreover, this method has the same reliability as the cryptographic certificate verification method as shown in [20].

A subject certificate can be another nested certificate. In this way, a nested certificate can be verified as the subject certificate of another nested certificate without using a cryptographic signature verification method.

2.5. Using nested certificates in PKIs and nested certificate paths

Nested certificates are not designed to replace all functions of the classical certificates, but to improve the performance and flexibility of them. Therefore, classical and nested certificates can be used together in PKIs and certificate paths. The PKI constructed in this way is called *Nested certificate based PKI* (*NPKI*). The certificates are verified via certificate paths in NPKI. The certificate paths that are extracted from an NPKI are called *nested certificate paths*. A *nested certificate path* is a chain of nested certificates together with a classical certificate at the end. This classical certificate is for the *target entity* of the nested certificate path. The eventual aim of using a nested certificate path is to certify the public key of the target entity. That is why the last certificate is a classical one. Each nested certificate of such a path, except the last nested certificate, is used to certify its subsequent nested certificate. The last nested certificate is to certify the classical certificate at the end.

A nested certificate path with *k* nested certificates is called *k-nested certificate path*. A generic *k*-nested certificate path (the certificates $nc_k,nc_{k\text{-}1},nc_{k\text{-}2}$ … nc_3,nc_2,nc_1,cc_0) is shown in Figure 4. In order to verify such a path, the verifier must obtain all certificates on it and must know the public key of A_k, the first NCA of the path. The verifier must also trust all CAs/NCAs on the path in order to verify the certificates issued by them. In a *k*-nested certificate path, each nested certificate is used to verify its subject certificate. At the end of a series of subject certificate verifications, the classical certificate, cc_0, of the target entity, *T*, is verified as the subject certificate of the last nested certificate, nc_1, of the *k*-nested certificate path. Only the first nested certificate, nc_k, of a *k*-nested certificate path is verified cryptographically using the public key of its issuer, A_k. The other certificates of the path are verified as the subject certificates. Therefore, the public keys of other nested and classical certificate issuers need not be found.

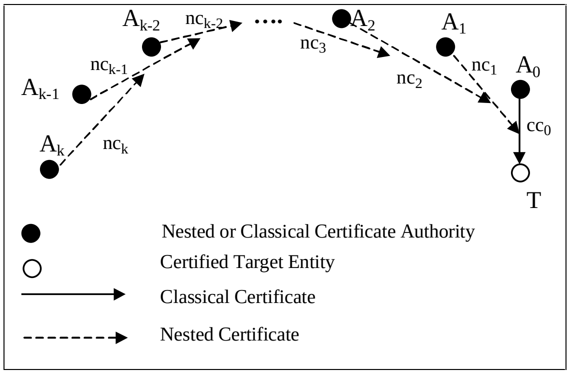

Figure 4. A generic *k*-nested certificate path

Since the subject certificate verification method is more efficient than the cryptographic certificate verification method, the use of nested certificates on certificate paths improves the nested certificate path verification time as compared to the classical certificate paths of the same length. The speed-up factor for nested certificate path verification method ranges between 1.9 and 8.8 for nested certificate paths with 1 to 8 nested certificates as discussed in [20].

An NPKI, which is based on free certification model, was proposed in [22]. In this model, every CA is free to choose classical or nested certificates to issue. There is no enforcement. Therefore, this model suggests an organic growth from zero.

The contribution of this paper is to propose another NPKI model, which is called the *transition from an existing PKI* model. This model aims to convert an existing hierarchical PKI into NPKI. It allows nested certification as well as classical certification. There is systematic nested certification enforcement in this model. Every CA issues nested certificates to the certificates that are issued by its neighbors in the PKI. In this way, it is possible to convert classical certificate paths into nested certificate paths without destroying existing hierarchical topology and trust relationships. The performance analysis shows that the use of nested certificates significantly improves the path verification time. There is a nested certificate issuance overhead for the authorities, but this overhead is acceptable in exchange for faster verification. The detailed performance evaluation of the proposed method will be given in Section 4.

3. NPKI construction using transition from an existing PKI approach

The construction of an NPKI using *transition from an existing PKI* approach deals with a systematic deployment of nested certificates throughout the infrastructure. Here, the goal is to have quickly verifiable nested certificate paths after this deployment. The methodology and characteristics of this approach are given in this section.

3.1. Transition from PKI to NPKI

There is systematic nested certification enforcement in this approach. The CAs behave as NCAs. The nested certificates are node-to-arc arcs in NPKI. The method for the transition is called *nested certificate propagation*. This method is examined in two steps: 1) *Setup*, 2) *End user addition/deletion/update*. The *setup* step addresses the actual transition from the existing PKI into NPKI. The *end user addition/deletion/update* step addresses the cases where a new user is added or an existing user is deleted or a user's public key is changed. The classical certificates of PKI are neither deleted nor disabled in NPKI. However, they are not used if there is a nested certificate path to replace the classical certificates. The classical certificates must be kept since they may be needed before transition is finished or when a new end user is added.

Our design choice is to work on tree shaped hierarchical PKIs to convert them into NPKI. Hierarchical topologies are, actually, the real world cases for most of the applications. However, there might be some cross certificates that destroy the pure hierarchy. Thoughts on cross certificates will be explained in Section 3.1.1.

3.1.1. Nested certificate propagation - setup

The common practice in classical PKIs is to verify the certificate of an end user starting with a CA/NCA. The aim of the setup step is to obtain nested certificate paths only towards the end users. These paths can start with any CA/NCA from which there exists a classical certificate path in the original PKI.

The basic rule behind the ability of the nested certificate issuance in PKI to form an NPKI is as follows. Let A be an authority and A^c be the set of authorities that have been certified by A. A can validate the certificates, which had been issued by the authorities in A^c, since A already knows the public keys of them. Consequently, A can issue nested certificates for all certificates (nested or classical) that had been issued by the authorities in A^c. The above condition is necessary but not a sufficient condition.

In our method, the CAs create nested certificates for all cases that are in conformity with the rule given above, but there is an exception: CAs do not issue nested certificates for the classical certificates which belong to other CAs. However, the classical certificates, which belong to the end users, are certified. All nested certificates, if possible, are also certified. In this way, a nested certificate path is produced for each classical certificate path from a CA to an end user.

In the setup step, nested certificate issuance propagates from leaf nodes (end users) towards the root. Actually, the leaf nodes and their parents do not issue any nested certificate. In the setup phase, first, nested certificates are produced by the CAs/NCAs, who are the grandparents of the leaf nodes. They issue nested certificates for the classical certificates of the leaf nodes. Then, these nested certificates are certified via other nested certificates issued by the parents of these CAs/NCAs. This propagating nested certificate issuance for nested certificates goes on until the root. At each iteration, each CA/NCA issues nested certificates for the nested certificates that are issued by its children.

An example setup phase over a tree-shaped PKI is given in Figure 5a and in Figure 5b. First, certificates of the end users are certified via nested certificates in Figure 5a. Then, these nested certificates are certified by the upper level root node in Figure 5b. As can be seen in Figure 5b, there is a nested certificate path towards each end user from all CAs/NCAs. However, as expected, there is no nested certificate path for the classical certificate paths between any two CAs/NCAs.

Any possible cross certificate that destroys the hierarchy is not certified via a nested certificate in order to keep the number of nested certificates at a reasonable level. The paths that use those cross certificates still use the same cross certificates. However, nested certificate paths can be used before and after the cross certificates.

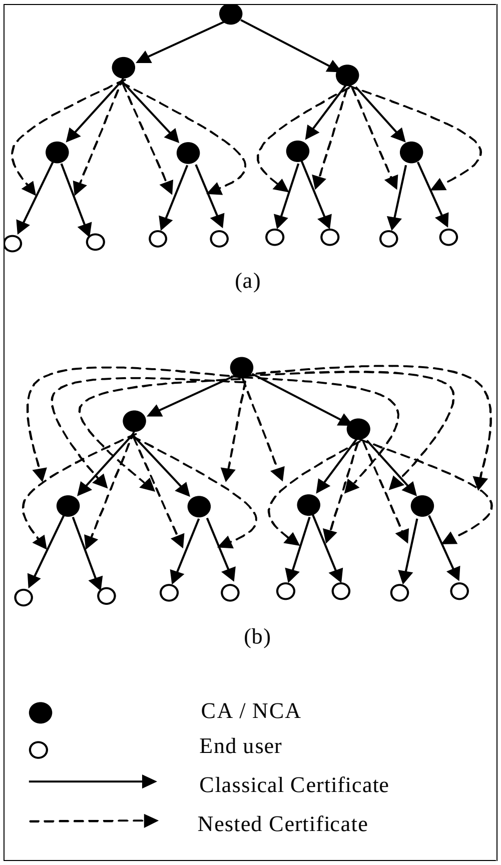

Figure 5. An example transition to NPKI in two steps

3.1.2. Nested certificate propagation – addition/deletion/update

A new user is added to a PKI via a classical certificate issued for it by a CA at upper-leaf level. The nested certificate propagation for this new user is the reduction of the setup algorithm for him/her. First, the classical certificate of the new user is certified by his/her grandparent CA/NCA with a nested certificate. Then, this nested certificate is certified via another nested certificate that is issued by the parent CA/NCA of the grandparent. This propagating nested certification goes on to the root such that, at each iteration, the nested certificate that is issued at the previous iteration is certified.

When a user is deleted, the classical certificate issued for this user is revoked. The revocation issues are discussed in Section 3.2.8.

To update the public key of a user, first its entry is deleted. Then, the user is added to the system as a new user with his/her new public key.

3.2. Characteristics of NPKI and the nested certificate propagation method

3.2.1. Efficient verification

The aim of the nested certificate propagation is to form nested certificate paths as alternative to the classical ones. The nested certificate path verification is faster than classical certificate path verification. These issues will be discussed in Section 4.

3.2.2. Dynamic certification

In classical PKIs, certification is a process between the issuer and the certified entity. However, certification affects several authorities in NPKI due to the nested certificate propagation method. This brings dynamism to the system such that the idle upper level authorities are involved in the lower level certification processes. As a result, the overall utilization increases. The nested certificate propagation method increases the load of the authorities but creates an opportunity to save time in the nested certificate path verification process. The time saved by the verifiers is more important than the time spent for nested certification as discussed in Section 4.4

3.2.3. Availability requirements of the authorities

In classical PKI, once a CA, *C*, issues a classical certificate for an end user, *E*, some classical certificate paths towards *E* are automatically created. However, in order to create nested certificate paths corresponding to these classical certificate paths, nested certificate propagation is necessary. At first glance, it seems that authorities need to be available on-line during the propagation process, but this is not correct. Although the authorities must issue nested certificates for propagation, they need not accomplish this task just after the classical certificate issuance for *E*. The propagation process may be completed in time. The nested certificate propagation is carried out only to form efficiently verifiable nested certificate paths towards *E*. However, the absence of some nested certificates on these paths does not cause non-verifiability of *E*, since there are compensating classical certificates on the PKI. Only the efficiency improvement of the nested certificate path verification is not fully utilized for these cases; but this is temporary.

It is sufficient for the NCAs to periodically check their neighbor CAs/NCAs in order to issue nested certificates for new certificates issued by their neighbors. The length of this period is related to the time for the completeness of the nested certificate propagation for *E*. If long time periods are chosen, the completion of the nested certificate propagation for *E* also takes a long time. However, even if the nested certificate propagation is not complete, each nested certificate issuance improves the average certificate path verification time.

Another important point about nested certificate propagation is that the propagation method is sequential, not parallel. This implies that, each authority should wait for its children to finish nested certification in order to issue nested certificates. Therefore, the propagation delay is cumulative.

3.2.4. Nested certificate propagation versus classical certificate propagation

An alternative approach to nested certificate propagation is the *classical certificate propagation* method, which produces direct classical certificates. Classical certificate propagation is theoretically possible in PGP [2] and ICE-TEL [7] systems. In this method, each CA, CA_1, verifies the paths towards the end users in the PKI starting with CA_1. Having verified each path, CA_1 issues a classical certificate for the verified end user. Using this method, it is possible to have only a single certificate between each CA - end user pair for which there was a classical certificate path in the PKI beforehand. This is an advantage of this method over nested certificate propagation. The average certificate path (actually, single certificate) verification time for classical certificate propagation method is less than average certificate path (actually, nested certificate path) verification time for the nested certificate propagation method. Moreover, the total number of nested certificates that must be issued in the nested certificate propagation method is equal to the total number of extra classical certificates that must be issued in the classical certificate propagation method. In addition, the total number of verifications that must be performed for propagation by each CA is the same in both methods. Why, then, must one insist on nested certificate propagation method, since it has no advantage over classical certificate propagation? Although the classical certificate propagation method is more advantageous, it is not possible to apply that method all the time. Some characteristics of the nested certificate propagation method make it preferable where classical certificate propagation is not possible:

1. Trust information-free certificate issuance.
2. Preservation of topology and trust relationships.

These characteristics will be discussed in Sections 3.2.5 and 3.2.6, respectively. The cases where classical certificate propagation is not possible are also addressed these sections.

3.2.5. Trust information-free certificate issuance

In order to verify a public key through a classical certificate path, all intermediate CAs on the path must be trusted. If even one of those CAs is not trusted, then the path cannot be verified. Therefore, in order to realize full classical certificate propagation, every CA of the PKI must trust everyone that can be reached starting with itself. Certainly, this is not a realistic assumption. On the other hand, partial classical certificate propagation is possible for the CAs, for which it is possible to verify some classical certificate paths that contain trusted CAs. The necessary condition for this case is that the direct classical certificate issuer must trust all intermediate CAs on the certificate path towards the target entity. However, if there is at least one untrusted CA on the path, then direct classical certificate issuer will not be able to verify the path and will not be able to issue a direct classical certificate for the target entity.

On the other hand, in order to issue a nested certificate for a subject certificate, the NCA does not need to trust anyone. Therefore, nested certificate propagation is applicable all the time, especially for the cases where classical certificate propagation is not applicable because of lack of trust. In order to issue a nested certificate, the NCA should only know the correct public key of the subject certificate issuer and should verify the signature over the subject certificate. The NCA does not need to trust the subject certificate issuer or the entity certified within the subject classical certificate, because by definition, a nested certificate does not guarantee the information correctness of the subject certificate content. Besides the verification of a nested certificate and its subject certificate via that nested certificate, the verifier must also trust the subject certificate issuer, in order to completely verify the subject certificate. In the nested certificate path verification, the verifier must trust all intermediate NCAs.

3.2.6. Preservation of topology and trust relationships

Classical certificate propagation method spoils the existing trust relationships in the PKI, since the CAs force the verifiers to by-pass the certificate paths by issuing direct classical certificates for the end users. Although this situation does not seem to be a problem at first glance, it may cause a problem for the strictly hierarchical PKIs, like the PKIs of DNSSEC [18], PEM [8] and SET [9]. In such strictly hierarchical PKIs, it is not possible to by-pass intermediate CA levels and consequently, classical certificate propagation is not possible for these PKIs.

For strictly hierarchical PKIs, trust relationships and hierarchical topology must be preserved. The authorities on PKI paths, which the verifiers trust, must be the same authorities after the propagation. Nested certificate propagation method preserves trust relationships and the topology in this manner as described below.

In the subject certificate verification method, the verifiers should trust both nested and subject certificate issuers in order to verify a subject certificate via a nested certificate. Moreover, in the nested certificate propagation method, each authority issues nested certificates for the certificates that have been issued by its neighbors in order to create one nested certificate path for each classical certificate path of the PKI. A nested certificate path created in this way passes through the CAs that its corresponding classical certificate path also passes through. As a consequence, in the nested certificate path verification process, the verifier must trust exactly the same CAs of the corresponding classical certificate path. In this way, trust relationships within the PKI are preserved in the NPKI, which is constructed via nested certificate propagation. Moreover, since the trust relationships are preserved and there is no level by-pass in the nested certificate propagation method, it can be claimed that the topology of the original PKI is also preserved in NPKI.

For example, consider Figure 6. In this figure, both a classical certificate path, which is the certificate sequence cc_6, cc_5, cc_4, cc_3, cc_2, cc_1, cc_0, and its corresponding nested certificate path, which is the certificate sequence nc_6, nc_5, nc_4, nc_3, nc_2, nc_1, cc_0, are shown together. The verifier has to trust the authorities $A_{0,}$ A_1, A_2, A_3, A_4, A_5 and A_6 in order to verify both classical and nested certificate paths. Moreover, in both classical and nested certificate path verifications, the verifier must know the public key of A_6 a priori.

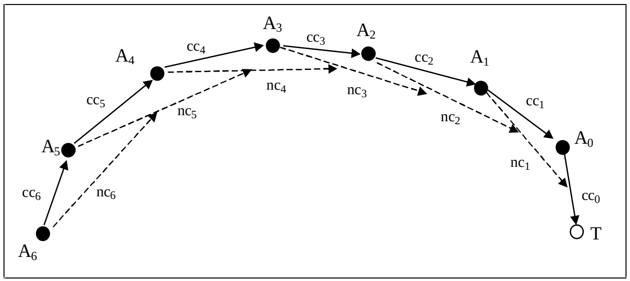

Figure 6. An example classical certificate path and its corresponding nested certificate path

3.2.7. Certificate storage and obtainment

The certificate storage and obtainment in NPKI are not different from classical PKIs. One solution uses distributed directories, while the other one has a centralized database. These solutions are explained below.

In NPKI, the certificates can be stored and obtained via distributed directories like X.500 directories [23] or the methods embedded in DNSSEC [18]. The CAs/NCAs may serve as directory servers or they may publish the certificates that they issue to other directory servers. The verifier queries the directory to get the certificates on a nested certificate path to verify the public key of a specific end user.

The objects stored in directories should belong to the entities with distinguishable names. The directory users query the directory to search for objects belonging to a specific name. Classical certificates are such objects, but nested certificates are not, because they are issued for other certificates with no distinguishable names. A possible solution to this problem is given below.

A common characteristic of nested certificates on a nested certificate path is that all of them can be used to verify only one classical certificate, which is at the end of the path. This classical certificate is to verify the public key of an end user. Therefore, the identity of this end user is the common attribute of all certificates on a nested certificate path, and it can be used as the distinguishable name for these certificates including the nested ones. However, since there can be several nested certificates on the path, the identity of the NCA of a nested certificate should also be considered for the sake of uniqueness. Moreover, the inclusion of the end user's identity in each nested certificate may facilitate the nested certificate path formation. The directory access mechanism can be improved so that the verifier is able to query the directory only once for all certificates containing the end user's identity. The returning certificates simply form a nested certificate path.

Another method of storing and obtaining certificates in NPKI may be to use a centralized database that contains all certificates in the system. In this method, as in the directory method, the nested certificates contain the identities of the end users of the corresponding nested certificate paths. Moreover, the database can be indexed by these end user identity fields. In that way, the retrieval process becomes faster and all certificates on a path can be obtained by a single query. However, as will be discussed in Section 4, the number of certificates in NPKI can be large. Therefore, having a single database may create storage capacity and corresponding efficiency problems.

3.2.8. Certificate revocation in NPKI

In a classical PKI, a classical certificate, *C*, is revoked if the private key corresponding to the public key in *C* or the private key of the CA issuing *C* is compromised. In

NPKI, the classical certificates for the end users are revoked under exactly the same conditions.

The ultimate aim of a nested certificate path is to verify the classical certificate at the end. Moreover, a nested certificate can take place on only one nested certificate path. Therefore, when a classical certificate is revoked for some reason, all nested certificates on the nested certificate path towards it automatically become useless. Consequently, these nested certificates need not be revoked.

Security policies generally require that nested certificates issued by an NCA whose key is compromised definitely not be used. Under these circumstances, all classical certificates at the end of the nested certificate paths, which contain a nested certificate issued by the compromised key, must be revoked. The nested certificates on these paths need not be revoked, since they automatically become useless. The best characteristic of this system is that the verifier needs to check the revocation status of only one certificate, independent of the path length. This certificate is the classical certificate at the end of the path; other nested certificates on the path need not be checked for revocation.

In the case of a hierarchical NPKI, the classical certificates that must be revoked are the ones that subordinate the NCA of the compromised key. Therefore, the compromise of a higher level NCA results in more revocations. This is, actually, the worst case scenario in classical PKIs too. Therefore, NPKI does not create an extra certificate revocation overhead as compared to worst case certificate revocation in classical PKIs. However, there may be lots of nested certificates that are not revoked but are useless. This situation inflates the databases/directories. A solution to this problem is to periodically run maintenance programs to locate and delete these useless nested certificates.

Revoked certificates may be kept in Certificate Revocation Lists (CRLs), as in the case of X.509. Each CA manages its own CRL. Certificate storage and obtainment methods can also be applied to CRLs.

The concepts that are given in this subsection deserve a more detailed examination. Formal analysis and the detailed performance evaluation of certificate revocation in NPKI are left as future research topics.

4. Performance evaluation

Nested certificate path verification is more efficient than classical certificate path verification. However, a significant number of nested certificates must be issued to convert a PKI into NPKI and there is a trade-off between the number of nested certificates and efficiency improvement in certificate path verification. This overhead is tolerable in order to have efficiently verifiable certificate paths. These performance analyses will be given in this section.

The PKI topology that will be analyzed in this section is a *k*-level *m*-ary balanced tree. In such a PKI, each non-leaf node issues *m* classical certificates for its child nodes and there are *k* non-leaf node levels. There is one node at level 0, which is the root CA. The leaf nodes are at level *k* and they represent the end users of the infrastructure. The non-leaf nodes are CAs of the PKI. They will also act as NCAs in the NPKI.

4.1. Performance analysis of nested certificate path verification

The analytical and simulation based performance evaluation of the nested certificate path verification method is given in [19], [20], [21]. The performance measure used in these analyses is the *speed-up* factor. The speed-up factor is the ratio of the classical certificate path verification time over the nested certificate path verification time. In [20], eight sets of simulations are performed; each uses a different pair of public-key cryptosystem (RSA [3] or DSA [6] with different key sizes) and hash algorithm (MD5 [4] or SHA-1 [5]). Paths with 1 to 8 nested certificates are considered in each set. Since there is one classical certificate at the end of each nested certificate path, there are 2 to 9 certificates total. The results for these simulations are shown in Figure 7. As can be seen from this figure, there is a remarkable improvement, especially for slower cryptosystems, like DSA-512 and RSA-2048. For the cases considered, the speed-up factors are between 1.9 and 8.8.

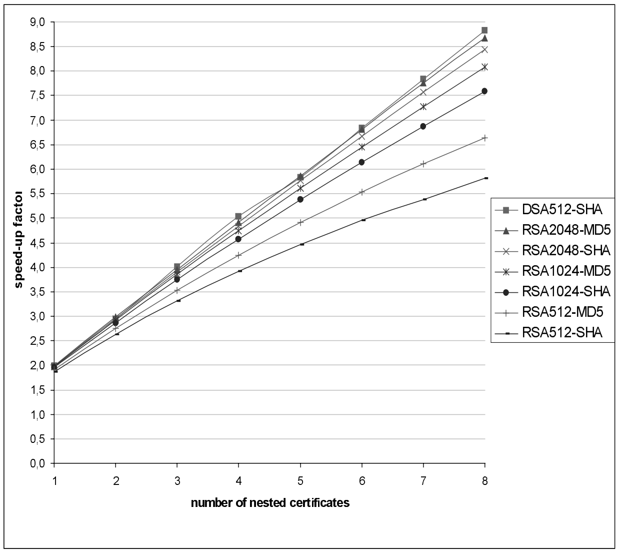

Figure 7. Change of speed-up factor with respect to the number of nested certificates on the nested certificate paths

Let *N* denote the number of nested certificates on the average length path of the produced NPKI. *N* is important for our analysis, since it is related to efficiency gain in nested certificate path verification. *N* is independent of *m* and equal to (*k* – 1)/2. Table 2 gives the *N* and corresponding nested certificate path verification speed-up factors for 3, 4 and 5 level trees. The speed-up factors are from Figure 7. As can be seen from Table 2, the efficiency improvement gets better as the number of levels increases.

Table 2. Speed-up factors for some hierarchical PKIs with different levels

Level (*k*)	**Number of nested certificates on average length path (*N*)**	**Speed-up factor range**
3	1	1.9 – 2.0
4	1.5	2.3 – 2.5
5	2	2.7 – 3.0

4.2. Nested certification overhead and trade-off analysis

Several nested certificates must be issued in the nested certificate propagation method. In this subsection, the Nested Certification Overhead to convert the whole PKI into NPKI, *NCO*, will be analyzed for a balanced tree shaped PKI/NPKI topology. *NCO* is the factor of increase in the total number of certificates by including the nested certificates in NPKI. This overhead value is always greater than or equal to 1. An overhead value of 1 means that there is no overhead. An overhead value of *x* means that nested certificate propagation increases the number of total certificates *x* times. Moreover, the trade-off between *NCO* and the performance improvement for the nested certificate path verification will be analyzed in this subsection. These analyses are performed using analytical methods; detailed formulation can be found in [19].

The change of *NCO* with respect to *m* is given in Figure 8 for 3, 4 and 5 level trees. The primary factor effecting *NCO* is the number of levels (*k*). The behavior of *NCO* is asymptotic and approaches to *k* as *m* increases. Therefore, the nested certification overhead cannot exceed the number of levels.

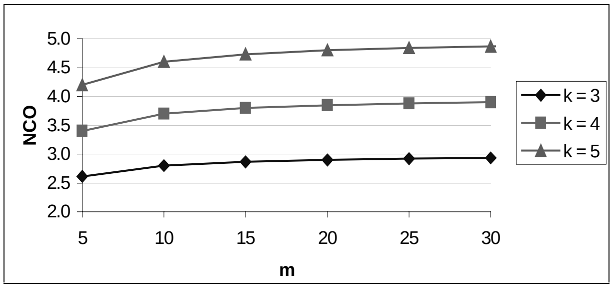

Figure 8. Change of *NCO* with respect to *m* for different levels

NCO is the primary disadvantage of the system. On the other hand, it is also possible to have nested certificate paths that can be verified more efficiently. Therefore, there is a trade-off between efficiency improvement and *NCO*. For example, let us consider a 4-level, 20-ary balanced tree PKI (i.e., *k* = 4, *m* = 20). The *NCO* value for this PKI is 3.85, while the nested certificate path verification speed-up factor is between 2.3 and 2.5. The trade-off is that the average path verification becomes 2.3 to 2.5 times faster by increasing the number of certificates 3.85 times. Although *NCO* is bigger than the efficiency improvement, this trade-off is acceptable, because the certificates are issued only once, but the verification can be performed several times.

Although *NCO* is low, the number of certificates is high. For the example PKI, the number classical certificates is 168,420. In order to carry out nested certificate propagation on this PKI, a total of 480,000 nested certificates must be produced. However, these nested certificates can be produced in a few hours.

4.3. Nested certification overhead distribution

The nested certification overhead is not evenly distributed among the NCAs in the hierarchy. However, the nested certification overhead is the same for the NCAs in the same level. The end users and their parents (levels *k* and *k*-1) do not issue any nested certificates in NPKI, so there is no nested certification overhead for them. For other levels, $i \in \{0 \ldots k-2\}$, the number of nested certificates that must be issued by an NCA in level *i* is given by m^{k-i}. That means, upper level NCAs (the ones with lower *i*) have to issue more nested certificates than the lower level NCAs (the ones with higher *i*).

Let us go back to our previous example of 4-level, 20-ary balanced tree PKI. The number of nested certificates issued by an NCA in each level is given in Table 3. As expected, the upper level NCAs produce more nested certificates than the lower level NCAs.

Table 3. Level by level number of nested certificates and number of NCAs for the 4-level 20-ary balanced tree PKI

Level (*i*)	**Number of nested certificates per NCA in the level (m^{k-i})**	**Number of NCAs in the level (m^i)**
0	160,000	1
1	8,000	20
2	400	400
3	0	8,000
4	0	160,000

4.4. Further discussion

An important criticism of the nested certificate propagation method may be the non-uniformity of the nested certificate issuance overhead. Indeed, a significant amount of nested certificates must be issued by upper level CAs of the NPKI, as shown in Table 3. In classical PKIs, the upper level CAs need not take any action, when a new end user is added to the PKI or the certificate for an end user is updated. However, in the nested certificate propagation method, when a new end user is added to the NPKI or the certificate for an end user is updated, one new nested certificate must be issued at each level. Although these characteristics seem to be disadvantageous, they can be justified:

1. Assuming that each CA/NCA stores the certificates that it has issued, the worst case (for the top level CA/NCA) storage requirement for the nested certificates is in the order of 10 Mbytes, which is quite acceptable.
2. The nested certificates are issued once, but they are used to verify nested certificate paths several times. Therefore, the overhead is once, but the gain is several times.
3. The classical certificates of the PKI still exist in the NPKI. Therefore, there are always the classical backups of the nested certificate paths in NPKI. Thus, nested certificate propagation can be considered as an “off-line” process. The authorities may issue nested certificates at the idle times. During the initial set up time or when a new end user certificate is issued, the verifiers may use the classical certificates until the nested certificate propagation is completed.
4. Classical CA servers are mostly idle and their utilization is small, since their classical certification loads are not so significant and they must be dedicated servers for security reasons. The utilization of these servers increases by the nested certificate propagation method.
5. Although the nested certificate issuance overhead seems to be significant, the example NPKI given above can be set up in a few hours without interfering with normal PKI operations. Such a setup time is acceptable. This setup time tends to increase for bigger PKIs and the nested certificate propagation method may become useless. For example, the set up time for a 5-level 30-ary balanced tree shaped PKI is 2-3 weeks. For such large PKIs, the bottleneck is mostly due to the top level CA, since it must issue large numbers of nested certificates itself. For these cases, nested certificate propagation can be considered for subhierarchies. That means, the top level CA does not issue nested certificates, but the CAs of its subhierarchies do. This approach can be recursive such that if the subhierarchies are also big, the nested certificate propagation is applied for their subhierarchies.

5. Conclusions and future work

Nested certification and the corresponding subject certificate verification methods were proposed to improve the certificate path verification times. A nested certificate is basically a certificate for another certificate. By using nested certificates in certificate paths, it is possible to have efficiently verifiable nested certificate paths. In this paper, the design of the *Nested certificate based PKI (NPKI)*, which incorporates both classical and nested certificates, has been presented. The NPKI construction model discussed in this paper is called *transition from existing PKI* and the method to realize the transition is called *nested certificate propagation* method. This model forces the CAs to issue nested certificates to the certificates, which are issued by their neighbor CAs in the PKI. The outcome of this model is a nested certificate path for each classical certificate path towards the end users in the PKI. On a nested certificate path, all certificates are nested certificates except the last one. Therefore, the verification time of a nested certificate path is considerably less than a classical certificate path. Another important advantage of the nested certificate propagation method is that the trust structure and the topology of the PKI are not spoiled. Therefore, NPKI preserves trust. Moreover, NPKI is dynamic since several authorities are involved in the certification of an end user in a distributed manner. In this way, the utilization of the authorities increases to the benefit of the verifiers. However, to attain such improvement, numerous nested certificates must be issued. That means, there is a trade-off between the verification improvement and the nested certificate issuance overhead. This trade-off was also analyzed in this paper by using a generic balanced tree PKI model. It has been observed that, for a 4-level, 20-ary balanced tree shaped PKI, the average path verification speed-up factor is between 2.3 and 2.5, depending on the cryptosystems and the hash algorithms used. However, the number of total certificates increases by 3.85 times. Unfortunately, this certification overhead is not distributed uniformly among the authorities. Upper level authorities perform more nested certifications than the lower level ones. However, these overheads are tolerable in order to improve the path verification time.

We point toward three areas of future work: (i) embedding of the NPKI into specific applications, e.g. e-mail, electronic commerce, electronic payment, etc., (ii) a more detailed analysis of the nested certificate revocation given in Section 3.2.8, (iii) the use of nested certificates for classical CRL signing.

Acknowledgments

This work was supported in part by TUBITAK (grant EEEAG-237). The authors are grateful to Audun Jøsang, Cetin Kaya Koc, John Burt and anonymous referees for their careful review and comments on this paper.

References

[1] ITU-T Recommendation X.509, ISO/IEC 9594-8, Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, 1997 Edition.

[2] Zimmermann, P., *PGP User's Guide*, available with free PGP software from http://www.pgpi.com, 1994.

[3] Rivest, R., A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, February 1978.

[4] Rivest, R., The MD5 Message Digest Algorithm, RFC 1321, April 1992.

[5] National Institute of Standards and Technology (NIST), Secure Hash Standard (SHS), Federal Information Processing Standard (FIPS) PUB 180 –1, U.S. Department of Commerce, Washington, 17 April 1995.

[6] National Institute of Standards and Technology (NIST), Federal Information Processing Standard (FIPS) PUB 186, Digital Signature Standard (DSS), U.S. Department of Commerce, 19 May 1994.

[7] Chadwick, D. W., A. J. Young, N. K. Cicovic, "Merging and Extending the PGP and PEM Trust Models – The ICE-TEL Trust Model," *IEEE Network*, vol. 11, no. 3, pp. 16-24, May/June 1997.

[8] Kent, S. T., "Internet Privacy Enhanced Mail," *Communications of the ACM*, vol. 36, no. 8, pp. 48-60, August 1993.

[9] MasterCard Inc., SET Secure Electronic Transaction Specification Book 1: Business Description, MasterCard Inc., May 1997.

[10] United States Postal Service, Performance Criteria for Information-based Indicia and Security Architecture for IBI Postage Metering Systems, August 1998, available from http://www.usps.gov/ibip/documents/specs/pc0819.pdf

[11] Housley, R., W. Ford, W. Polk and D. Solo, Internet Public Key Infrastructure: X.509 Certificate and CRL Profile, RFC 2459, March 1999.

[12] Adams, C. and S. Farrell, Internet X.509 Public Key Infrastructure Certificate Management Protocols, RFC 2510, March 1999.

[13] Ramsdell, B., S/MIME Version 3 Certificate Handling, work in progress, Internet Draft <draft-ietf-smime-cert-08.txt>, April 1999.

[14] Chokhani, S., "Towards a National Public Key Infrastructure," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 70-74, September 1994.

[15] Ellison, C. M., et. al., SPKI Certificate Theory, work in progress, Internet Draft < draft-ietf-spki-cert-theory-04.txt>, 17 November 1998.

[16] Ellison, C. M., et. al., Simple Public Key Certificate, work in progress, Internet Draft <draft-ietf-spki-cert-structure-05.txt>, 13 March 1998.

[17] Rivest, R. and B. Lampson, "SDSI – A Simple Distributed Security Infrastructure", 1996, http://theory.lcs.mit.edu/~cis/sdsi.html

[18] Eastlake, D. and C. Kaufman, Domain Name System Security Extensions, RFC 2065, January 1997.

[19] Levi, A., "Design and Performance Evaluation of the Nested Certification Scheme and its Applications in Public Key Infrastructures", Ph.D. Thesis, Bogazici University, Dept. of Computer Engineering, May 1999.

[20] Levi, A. and M. U. Caglayan, "Verification of Classical Certificates via Nested Certificates and Nested Certificate Paths," *Proceedings of ICCCN99 – Eight International Conference on Computer Communications and Networks*, pp. 242 – 247, 11 – 13 October, 1999, Boston, Massachusetts.

[21] Levi, A. and M. U. Caglayan, "Analytical Performance Evaluation of Nested Certificates," *Performance Evaluation*, vol. 36-37, pp. 213 - 232, August 1999.

[22] Levi, A. and M. U. Caglayan, "NPKI: Nested Certificate Based Public Key Infrastructure," *Advances in Computer and Information Sciences '98 - Proceedings of the Thirteenth International Symposium on Computer and Information Sciences - ISCIS XIII*, IOS Press, Concurrent Systems Engineering Series, vol. 53, pp. 397 – 404, October 1998, Turkey.

[23] ITU-T Recommendation X.500, ISO/IEC 9594-1, Information Technology - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Services, 1997 Edition.