NPKI: Nested Certificate Based Public Key Infrastructure[1]

Albert Levi and M. Ufuk Çağlayan
Boğaziçi University, Department of Computer Engineering,
Bebek, Istanbul 80815, Turkey
levi@boun.edu.tr *caglayan@boun.edu.tr*

Abstract: X.509 based Public Key Infrastructures use classical certificates. To verify a classical certificate the public key of the issuer Certificate Authority (CA) must be known and that CA must be trusted. The nested certificates have been proposed to relax the trust requirements of certificate issuance and to by-pass the necessity of public key information to verify a classical certificate. This paper describes the basic principles and the certificate path verification scheme of a Nested certificate based Public Key Infrastructure (NPKI). In NPKI, the classical certificates are used together with the nested certificates and it is a generic system. The basic advantage of NPKI is the ability to connect the disconnected classical certificate paths by using nested certificates and by this way, the verifiers can form alternative certificate paths that cannot be formed using only the classical certificates. Moreover, in NPKI, the authorities are more flexible than the authorities of the classical PKIs, since they can issue nested certificates in situations where the classical certificates cannot be issued.

1. Introduction

As the Internet becomes a global media for electronic data interchange, the security related problems have been brought out as a handicap for development of Internet applications like e-mail, home banking, electronic commerce and business. One of the important problems is the identity proving problem which is critical especially for the financial transactions and applications. Identity proving problem can be defined as the problem of ensuring the real identity of a message or transaction originator.

The most common methods used for the solution of the identity proving problem are public key based *certificate* systems. A *certificate* is a signed binding between the public key and the identity of an entity. The most common certificate scheme is the ISO/ITU X.509 [5] recommendation. The reader may refer to [1] and [5] for more information on certificates. The certificates are signed by trusted Certificate Authorities (CAs). Moreover, there must be a network to connect the individual CAs. By this way, the network entities who have certificates from different CAs will be able to verify each other's certificate. The network of CAs is named as the *Public Key Infrastructure* (*PKI*). A PKI is a directed graph in which arcs represent the certificates from the CAs towards the certified entities. Although tree like hierarchies are preferred in some major PKIs like SET [6] and PEM [3], mesh topologies are also possible using cross certificates. Actually, the X.509 [5] standard does not enforce any standard topology for the PKIs, because such topologies are application oriented and must be defined by the applications.

[1] This work has been supported by Boğaziçi University Research Fund under grant 97A0102 and by State Planning Organisation (DPT) under grant 96K120490.

In order to verify a certificate and to find out the public key of the target entity, the verifier must know the correct public key of the CA of the target certificate. If the verifier does not know, then she has to verify the certificate for that CA and to do so she has to know the public key of the CA of the CA of the target entity. This loop goes on until the verifier is faced with a certificate that she can directly verify (i.e. she knows the public key of the corresponding CA). The certificates in that loop constitute a path which is called the *certificate path*. This path is a directed one and the starting point is a CA that the verifier knows the public key and the ending point is the target. The most important point here is the fact that each entity of the path has to issue a certificate for the next entity so that series certificate verifications would be possible. The certificates of such a path are drawn from the PKI.

The *Nested Certificates* [2, 7] are proposed as an alternative certification scheme. Classical certificates give assurance about the identity - public key binding of an entity, whereas a nested certificate certifies another certificate. The certified certificate is called *subject certificate*. By issuing a nested certificate, the issuer of that certificate guarantees that the subject certificate has been signed by the claimed authority, but does not guarantee the correctness of the information in the subject certificate content. The nested certificates give less assurance and they can be issued with limited trust information. Therefore, the nested certificates can be issued where the classical certificates cannot. On the other hand, the verifiers can verify a classical certificate via a nested certificate without using the public key of the classical certificate issuer.

In this paper, the principles of a novel model, in which the nested certificates are adapted to PKI schemes, are presented and discussed. The model is called *NPKI* (*Nested certificate based PKI*). In NPKI, both X.509 based classical certificates and the nested certificates can be used. The PKI model of NPKI is a multi-purpose, general PKI and is aimed to be used by any application which requires public key based digital certificates. The use of the nested certificates in the NPKI makes the system more flexible in both certification and the verification points of view. The usage of the nested certificates will allow some extra certification relationships that cannot be achieved using the classical certificates in a PKI. By this way, a classical certificate can be certified by a nested certificate. Consequently, the verifiers will be able to constitute certificate paths in which the disconnected classical certificate paths are connected via nested certificates.

In the next subsection, the general characteristics of the NPKI will be explained. Section 2 will describe the nested certificate concept. In Section 3, the general structure, advantages, certificate paths and certificate path verification algorithm of the NPKI will be detailed. Section 4 contains the conclusions and the future work.

1.1. The general characteristics of NPKI

In this paper, we will give the principles of a novel system that imports the nested certificates into a generic PKI. This system is called *Nested certificate based PKI* (*NPKI*). In this subsection, we will list some basic design criteria and assumptions of NPKI. Those assumptions are general assumptions which are valid for the PKI part of the NPKI. The rules and assumptions to import the nested certificates into NPKI will be explained in the Sections 3 and 4.

The design criteria of NPKI are to be *generic*, *flexible* and to have a *user oriented trust-management*. Those criteria are detailed below:

Generic: The NPKI is not aimed to work for a specific application like E-mail, electronic payment. Instead, it is designed to show the results and the advantages of adding the nested certificate concept to the classical PKIs. That is why a generic basic design has been preferred and application specific design decisions have been avoided. The NPKI can be

considered as a framework to adapt the nested certificates into PKIs. We believe that specific NPKI designs for different applications will follow this paper. Secure and private E-mail, electronic commerce and payment are the most suitable systems to have a NPKI.

Flexible: NPKI is a general system. Therefore, it should be flexible in order to port it to a specific application. Since no application specific design decision has been assumed in NPKI, it can be easily adapted to a specific application. Moreover, the aim of the nested certificate usage in the NPKI is to provide more flexible certification and verification structure.

User oriented trust management: The key component of the NPKI is the user. The CAs, certified objects and the verifiers are the users of NPKI. NPKI will be a X.509 based PKI. Because of the regulative behaviour of the standard X.509 certificate structure, the users are retained to have a high degree of freedom and flexibility to construct their own trust policies. The NPKI will allow the highest degree of freedom, flexibility and easiness that X.509 allows. For example, to attain that goal the NPKI will use policy identifiers and cross certificates [5]. Furthermore, the users of NPKI will have extra flexibility to constitute their own trust and certification policies because of the use of nested certificates.

The general design assumptions of NPKI to satisfy the above criteria are as follows:

Certificate Structure: In NPKI, the certificate structure conforms to X.509 standard. X.509 is the only international standard in the era of digital certificates and the new products which uses digital certificates are getting more and more X.509 compliant.

Topology: There is no restriction for the topology of the network of certificates in NPKI. It can be a tree or a web based directed graph structure. The web structure (if any) can be constructed using cross certificates or by having possessed several certificates from different CAs. In the design of NPKI, the cross certificates will not be differentiated from classical certificates.

Trust and certification path processing: In NPKI, the users will have a partial freedom to choose the trusted CA groups via policy identifiers [5] that X.509 allows. Moreover, the users will be able to specify their trusted CAs with the corresponding public keys, in their local databases. By this way, the users will be able to specify their directly trusted starting CAs of the certification paths. The standard X.509 certificate path processing rules [5] will be used in NPKI.

Network address types: Since NPKI is not for a specific application, there is no need to restrict the type of network address. Thus, in NPKI the network address specified in a certificate can be any address (e-mail, URL, etc.) that the X.509 standard allows.

Who signs the certificate? In the literature, there is a discussion about the executive object for the digital signatures. Some researchers say that the digital signatures are issued by the keys, while the others argue that the signatures are issued by individuals who own the key. Reiter and Stubblebine [4] have given a good discussion of this subject and some other design criteria for the PKIs. The widely accepted answer to this question is to have the keys as the executive signing object for digital signatures and we will accept this answer for NPKI. Therefore, we will try to validate the keys, rather than the owners, within the certificates as the signer of the next certificate in the certificate path.

Certificate Revocation: We will not deal with certificate revocation for NPKI in this paper and leave that topic as a further research orientation.

2. Nested Certificates

The nested certificates are proposed by Levi and Çağlayan in [2, 7]. In this section, the general characteristics of nested certificates, certificate verification and the advantages of

nested certificates will be summarised. We encourage the readers to refer [2, 7] for more information.

2.1. Definitions

Classical certificate: A X.509 certificate that has been explained in [5].
Nested Certifier / Nested Certificate Authority (NCA): Issuer for a nested certificate.
Nested Certificate: A certificate that certifies another (subject) certificate. By issuing a nested certificate, the NCA assures that the signature over the subject certificate is authentic and legitimate, but does not guarantee the correctness of the information within the subject certificate content. That means a nested certificate does not certify the correct public key of the entity certified by its subject certificate.
Hierarchical nested certificate: A nested certificate for which the subject certificate is another nested certificate.

2.2. Advantages of Nested Certificates

2.2.1. Certificate Issuance with Limited Trust Information

A NCA may easily issue a classical certificate for an entity. However, a nested certificate gives less assurance than a classical certificate. Therefore, a NCA may prefer to issue a nested certificate instead of a classical certificate. Moreover, nested certificates can be issued with limited trust information, so they can be issued where the classical certificates cannot. For example, suppose that the entity has a classical certificate from another CA and the NCA knows the public key of that CA, so she can easily verify that certificate. Also assume that, she does not trust that CA as an introducer. The NCA is in a position that she knows that the signature over the certificate is valid (because, she knows the public key of the CA), but the certificate content is suspicious. Therefore, NCA can not issue a classical certificate for the entity, but she can issue a nested certificate for the classical certificate. Because, she knows that the signature for the classical certificate is valid and she does not guarantee the correctness of it in a nested certificate.

2.2.2. Certificate Verification Without Using Public Key

The verifier can verify a classical certificate if the following two conditions are satisfied: i) the verifier must know the correct public of the CA for the classical certificate and ii) that CA must be trusted.

If one of the above conditions is not met then, the verifier can not verify the certificate. Let us suppose that the verifier trusts the CA for the classical certificate, but does not have the corresponding public key to verify it, and also suppose that there is a nested certificate over that classical certificate. Let us also assume that the verifier trusts the NCA for the nested certificate and knows the public key of the NCA. Under these circumstances, the verifier can verify the nested certificate and consequently learn the signature over the classical certificate. Then, the verifier compares that signature with the actual signature over the classical certificate. If they are the same, then the verifier verifies the classical certificate and validates the public key within that classical certificate, since the CA for the classical certificate is trusted. From the above scenario it can be easily seen that, the verifier has verified the classical certificate without using the public key of the CA of that certificate. This is actually the most important advantage of the nested certificates.

Besides the validity of the signature over a classical certificate, the verifier should also check whether that classical certificate is revoked or not. Certificate revocation is an expensive process and generally maintained by Certificate Revocation Lists (CRLs). A detailed explanation of certificate revocation can be found in [5]. By revocation of a certificate, the public key within that certificate becomes invalid. Therefore, the certificates,

which had been issued by this invalid key, could not be verified after the revocation time. However, if a classical certificate, which had been issued by a revoked key, has a nested certificate, then that classical certificate could be verified. Because, in this case the verifier does not need to know the public key of the classical certificate issuer to verify the classical certificate and consequently, the invalidity of this public key does not cause any problem.

3. The NPKI

In this section, the general structure of NPKI, the design considerations, advantages and the certificate path processing in the NPKI will be detailed.

3.1. General Structure of NPKI

The NPKI employs both the classical certificates and the nested certificates. Actually, NPKI is a network of those types of certificates. For the sake of simplicity, the cross certificates are considered as classical certificates in NPKI. The nodes of the NPKI network are the end users, CAs and NCAs. In NPKI, the end users are distinguished from CAs and NCAs, since X.509 certificate structure enforces such discrimination. However, the CAs and the NCAs can be the same entities. Therefore, the CAs and the NCAs are not differentiated in NPKI.

The arcs for this network will be the certificates. A classical certificate is a certificate from a CA to an end user or to another CA, so it is a node-to-node arc in the representation. On the other hand, a nested certificate is a certificate from a NCA to another certificate, therefore a nested certificate is an arc from a node to another arc in the representation. Moreover, the NPKI also supports hierarchical nested certificates which are explained in Section 2.1. A hierarchical nested certificate is a nested certificate for another nested certificate and its representation is again an arc from a node to another arc. However, the subject arc for a hierarchical nested certificate is another nested certificate. An example NPKI network is depicted in Figure 1. The hierarchical and normal nested certificates do not have different structures. Therefore, they have not shown differently in the Figure 1.

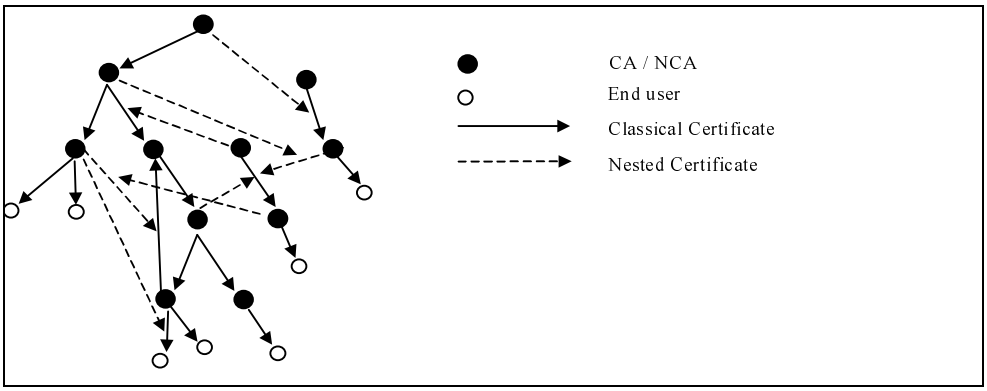

Figure 1 An example NPKI network

3.2. Assumptions

The general design assumptions of NPKI were given in Section 1.1. In this subsection the assumptions about the nested certificates and the NCAs will be explained.

A new type of certificate: The content and the usage of a nested certificate are different from a classical certificate. Thus, in NPKI the nested certificates are considered as a new certificate type.

CAs and NCAs are the same entities: Although the nested certificates are different from the classical certificates, they can be issued by the same authorities. Therefore, the CAs and NCAs can be the same NPKI entities, but an end user cannot be a CA or NCA.
Trust Considerations for NCAs: An entity can issue a classical certificate as well as a nested certificate, because of the above assumption. However, a user may trust an entity differently as a CA and as a NCA. Therefore, trust to a CA and trust to a NCA need to be interpreted differently. A discussion about these different trust interpretations for the CAs and the NCAs can be found in [2].

The X.509 standard has not been designed for certificates other than the classical certificates. Therefore, the implementation of NPKI with the above assumptions in compliance with X.509 may require some minor modifications and interpretation changes in the X.509 standard. Those modifications and changes are discussed briefly in [7].

3.3. Certificate Path Processing in NPKI

In NPKI, in order to verify the binding between the identity and the correct public key for a target entity, the verifier must find a valid certificate path for which the end point is the target. The starting point of a certificate path is either the verifier itself or another CA or NCA to whom the verifier directly trusts. To have a CA or NCA other than the verifier as the starting point of a certificate path is allowed only if the verifier has an entry for that CA or NCA in her local *trusted authorities database*. The verifier has to know also the correct public key of the first entity of the certificate path. Moreover, the last certificate of that path must be a classical certificate, because that certificate is used to verify the public key of the target and only the classical certificates can certify public keys. Other certificates of a certificate path may be either classical or nested certificates. Each classical certificate is verified to find the public key of the CA or the NCA of the next certificate in turn. On the other hand, the nested certificates of the certificate path are verified to validate their subject certificates. All of the certificates of a certificate path until the target entity are processed one by one sequentially. All of the intermediate certificates must be verified successfully, in order to consider the certificate path as valid and consequently validate the public key of the target entity.

The algorithm for certificate path processing for NPKI is given in Figure 2. That algorithm explains the general principles of certificate processing conceptually. Actual implementation requires extra data structures, variables and predefined operations. For example, to check if an authority is trusted or not, the verifier checks whether the policy identifier within the certificate is in her allowed policy identifiers set or not. If so, she trusts the authority, otherwise she does not trust. Similarly, another predefined operation is necessary to check the validity of a signature over a certificate. All the implementation details are defined for classical certificates in X.509 [5]. X.509 conformance issues of the nested certificates are discussed briefly in [7]. The below algorithm is for a single path. In the NPKI there may be several paths between any two entities. If one path fails to validate the target entity, then other paths should be tried.

3.4. Advantages of NPKI

An authority may prefer to issue a nested certificate instead of a classical one, since the assurance of a nested certificate is less than a classical certificate and it can be issued in case of limited trust information. Therefore, the authorities of NPKI are more flexible. In NPKI, even if the public key of a CA was not known or revoked, the certificates issued by that CA could be validated by using the nested certificates. By this way the verifiers can combine disconnected classical certificate paths via nested certificates and form alternative

certificate paths that cannot be constructed using only classical certificates as shown in Figure 3. Such combinations can be done by using a single nested certificate (the certificate 2 in Figure 3) or by a chain of nested certificates (the certificates 5,6 in Figure 3).

Verification of a classical certificate via a chain of nested certificates is more efficient than the verification via a chain of classical certificates. Because, all of the classical certificates must be verified cryptographically and cryptographic public key signature verification is an inefficient process. On the other hand, only the first nested certificate of a nested certificate chain must be verified cryptographically, the other nested certificates of the chain are verified simply by signature comparisons. A detailed discussion on the performance analysis of the nested certificate verification can be found in [7].

```
CertPathGood ← TRUE;
FOR CertInTurn := the first certificate of the path TO the last certificate DO
    IF CertInTurn is a classical certificate THEN
[verification of a classical certificate via a nested certificate]
        IF the certificate of the previous iteration was a nested certificate
        THEN
            IF the issuer of CertInTurn is not trusted for the verifier as CA
            OR the signature that had been verified in the previous iteration
            is not the same as the actual signature over CertInTurn THEN
                CertPathGood ← FALSE
            ELSE
                "CertInTurn has been verified and the public key within
                CertInTurn has been validated"
[Cryptographic verification of a classical certificate]
        ELSE
            IF the issuer of CertInTurn is not trusted for the verifier as a CA
            OR the signature over CertInTurn is invalid THEN
                CertPathGood ← FALSE
            ELSE
                "CertInTurn has been verified and the public within it has been
                validated"
    ELSE
        IF the CertInTurn is a nested certificate THEN
[verification of a nested certificate via another nested certificate]
            IF the certificate of the previous iteration was a nested
            certificate THEN
                IF the issuer for CertInTurn is not trusted for the verifier as
                a NCA OR the signature that had been verified in the previous
                iteration is not the same as the actual signature over CertInTurn
                THEN
                    CertPathGood ← FALSE
                ELSE
                    "CertInTurn and the subject signature (of the subject
                    certificate) that it certified have been verified"
[Cryptographic verification of a nested certificate]
            ELSE
                IF the issuer of CertInTurn is not trusted for the verifier as a
                NCA OR the signature over CertInTurn is invalid THEN
                    CertPathGood ← FALSE
                ELSE
                    "CertInTurn and the subject signature (of the subject
                    certificate) that it certified have been verified";
IF CertPathGood THEN
  RETURN "The Public Key of the Target has been verified successfully"
ELSE
  RETURN "The Certificate Path is Invalid, the Public Key of the Target has not
  been verified"
```

Figure 2 Algorithm for Certificate Path Processing in NPKI

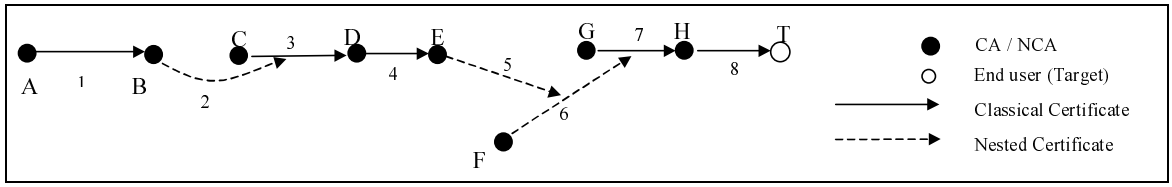

Figure 3 An example Certificate Path for NPKI

4. Conclusions and Future Work

In this paper, the design of NPKI, a X.509 based PKI which incorporates the nested certificates as well, has been presented. The nested certificates [2, 7] are used to realise flexible certification and verification scenarios. By using the nested certificates in NPKI, disconnected classical certificate paths can be connected and different advantageous certificate path alternatives can be derived. Moreover, the certification authorities of the NPKI is more flexible than the ones of other PKIs, because they have the alternative of issuing a nested certificate for the cases that they cannot issue a classical certificate. Moreover, using the nested certificates in certificate paths improves the efficiency of certificate verification [7]. Another important point which is explained in [7] is that the usage of nested certificates in NPKI does not cause important incompatibilities with the X.509 certificate structure and the nested certificates can be implemented as X.509 certificates with minor modifications.

As a further research direction, the design of NPKI can be extended to detail the nested certificate issuance policies. Moreover, we did not deal with the revocation of the nested certificates in this paper. Although we believe that the nested certificate revocation is not so different from the revocation of classical certificates, nested certificate revocation and using nested certificates to sign the CRLs can be examined as other future research topics.

The NPKI is general. As another further research area, the concepts of NPKI can be adapted to specific applications (like E-mail, electronic commerce, electronic payment) to use the nested certificates in their PKIs.

The nested certificates can also be interpreted as the transfer of the liabilities and responsibilities of the CAs to the NCAs. The reasoning behind this interpretation is the fact that, the object signed in a nested certificate is the signature of the CA and by signing a signature, the second signer accepts the conditions that the first signer had accepted. This interpretation can be used in the PKI applications for which the CAs have some responsibilities like the model due to Reiter and Stubblebine [4]. In this model, the CAs insure the name to key bindings of the users by issuing a certificate. Therefore, the CAs are liable to pay the insured amount in case of a misbehaviour of a certified user. By adding the nested certificate concept in this model, the liability of the CAs can be transferred to some other parties. This would be useful to add a reassurance component to Reiter and Stubblebine model. We will not use the nested certificates for liability transfer in NPKI, since the way of using this interpretation is application specific.

References

[1] W. Stallings, Network and Internetwork Security, Prentice Hall, 1995.

[2] A. Levi, M. U. Çağlayan, A Multiple Signature Based Certificate Verification Scheme, in the proceedings of BAS'98, The Third Symposium on Computer Networks, 25-26 June 1998, İzmir, Türkiye, pp. 1 -10

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

[4] M. K. Reiter, S. G. Stubblebine, Toward Acceptable Metrics of Authentication, in the Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp 10 –20, May 1997, Oakland CA.

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

[6] Secure Electronic Transaction (SET) Homepage, `http://www.mastercard.com/set/`

[7] A. Levi, M. U. Çağlayan, Nested Certificates and Their Applications in Public Key Infrastructures, Technical Report FBE/CmpE-02/98-13, Boğaziçi University, 1998.