Verification of Classical Certificates via Nested Certificates and Nested Certificate Paths

Albert Levi and M. Ufuk Caglayan
Bogazici University, Department of Computer Engineering, Bebek, 80815 Istanbul, Turkey

***Abstract* - Nested certificates are used to certify their subject certificates. In this way, the subject certificates can be verified via their nested certificates without using signature verification methods based on public key cryptosystems. Such a verification method is called as subject certificate verification. In this paper, subject certificate verification method will be introduced. It will be shown that subject certificate verification has the same confidence as the cryptographic certificate verification also. Moreover, subject certificate verification is faster than the cryptographic certificate verification. It will also be shown that a classical certificate can be verified via a sequence of nested certificates – called nested certificate path – and such verification has the same confidence as the cryptographic verification of the same certificate. Nested certificate path verification is faster than the classical certificate path verification also. Moreover in this paper, simulation results will be presented for the efficiency improvement in the nested certificate path verification method over the cryptographic classical certificate path verification method.**

I. Introduction and Background Information

Public key cryptography based systems are popularly used in the network security and authentication applications. One of the basic problems of these systems is to find out the correct public keys of the entities. The digital certification mechanism is the widely accepted mechanism for the solution to this problem. The International Telecommunications Union has proposed the X.509 [1] standard for the use of digital certificates. The X.509 certificates are used to verify the binding between the identity and the public key of the entities. The certificates are issued by trusted *Certification Authorities* (*CAs*). In order to find out the correct public key of a target entity, the verifier must follow a *certificate path* with several certificates. In this path, each certificate is verified to find out the public key of the next CA and each public key is used to verify the next certificate. Throughout the paper, the terms of X.509 *certificates* and *classical certificates*, and the terms of *X.509 certificate path* and *classical certificate path* will be used interchangeably.

A. Nested Certificates

Nested Certification [2] is a new type of certification. The certificates issued in this scheme are called *nested certificates*. Nested certificates are used to guarantee the integrity and correctness of the signature over a *subject* certificate. Therefore, a nested certificate is considered as a certificate for another certificate. The basic difference between a nested certificate and a classical one is that a nested certificate is to certify its subject certificate, whereas a classical certificate is to certify a public key. The subject certificates can be classical certificates or other nested certificates. In this way, it is possible to certify both classical certificates and other nested certificates via nested certificates.

The nested certificate issuance is similar to classical certificate issuance. The nested certificates are issued by the digital signature of the *Nested Certificate Authority* (*NCA*) over the nested certificate content. The content of a nested certificate is related to its requirements. The two requirements of a nested certificate are: 1) to certify that the subject certificate content has been signed by the claimed CA or NCA and 2) to certify that the subject certificate content has not been maliciously modified.

In order to satisfy the first requirement, a nested certificate contains the existing signature over the subject certificate content. In order to satisfy the second requirement, a nested certificate contains the hash of its subject certificate content. The hash of the subject certificate content can be obtained by applying an irreversible one way hash function [3, 4] to the subject certificate content. The structure of a nested certificate and its relationships with the corresponding subject certificate are given in Fig. 1.

By issuing a nested certificate, the NCA assures that the subject certificate has been signed by the claimed issuer of the subject certificate and has not been modified maliciously. In order to issue a nested certificate, the NCA of the certifier nested certificate must have verified the signature over the subject certificate content successfully.

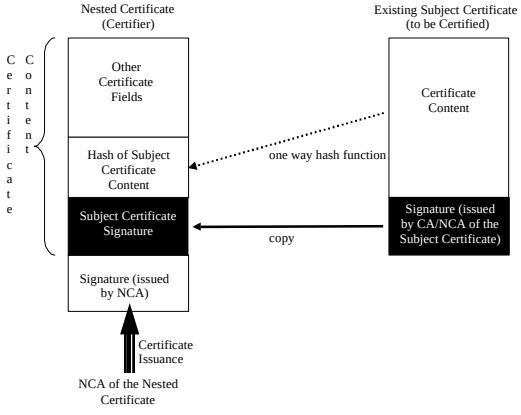

Fig. 1. The structure of a nested certificate

This work was supported in part by Turkish State Planning Organization (grant 96K120490), Bogazici University Research Fund (grant 97A0102) and TUBITAK (grant EEEAG-237).

B. Cryptographic Certificate and Classical Certificate Path Verification

The classical verification method for the classical certificates is *cryptographic verification*. In this method, the digital signature over the classical certificate content is verified by employing public key cryptosystem based signature verification operations. Cryptographic verification method can also be applied to nested certificate verification. The basic rule of the cryptographic certificate verification is the existence of a legitimate digital signature, which had been issued by a trusted certificate issuer, over the certificate content. The trust to the certificate issuer is determined by using policy identifiers in X.509 [1]. The digital signatures are issued and verified by employing a public key cryptosystem, like the RSA [5] or the DSA [6] cryptosystems. First, the certificate issuer calculates the hash of the certificate content using a one way hash algorithm, like the MD5 [3] or the SHA-1 [4] algorithms. Then, the issuer signs that hash in order to issue a certificate.

In order to cryptographically verify a classical or a nested certificate, *cert*, the verifier must know the correct public key of the issuer of *cert*. Assuming that this public key is known by the verifier, to cryptographically verify *cert*:

1. The verifier first applies the one way hash algorithm to the content of *cert*.

2. The verifier applies the public key cryptosystem based signature verification procedure to the signature part of *cert* using the public key of the issuer of *cert*.

3. The verifier compares the outcomes of above steps. If they are the same, then the verifier makes sure about: (i) the integrity of the content of *cert* and (ii) the legitimacy of the signature of the issuer of *cert* over the content of *cert*.

4. If the issuer of *cert* is trusted by the verifier, then the above two results of the verification imply that: (iii) the information given in *cert* is correct.

This verification algorithm is valid for the verification of both classical and nested certificates. However, since the contents of the classical and nested certificates are different, the implications of the verification of these two types of certificates are also different. In other words, the information given in *cert* is interpreted differently in classical and nested certificate cases. Verification of a classical certificate yields information about the legitimacy of the public key within the classical certificate. On the other hand, verification of a nested certificate returns information about the correct hash value and signature over its subject certificate content.

Classical certificate path is a chain of classical certificates. They are formed to verify the correctness of the public key of a target entity *T*. The verifier *V* verifies all of the certificates one by one sequentially. The verification starts with the first certificate of the path and *V* must know the correct public key of the first CA. However, *V* may not be the first CA, it may be any user. On the other hand, the trust of *V* to all of the CAs on the path is essential in order to verify the path. The cryptographic certificate verification steps are applied for the verification of all of the certificates on the path. Each certificate verification yields a public key, which is to be used to verify the next certificate on the path. This iterative process goes on until the target entity is reached. In order to verify the public key of the target entity, all of the certificate signatures must be legitimate and all of the CAs must be trusted by the verifier.

C. Contribution of the Paper

A significant overhead of classical certificate path verification is the cryptographic verification necessity of all the certificates on the path. The public key cryptosystem operations are computationally complex, therefore inefficient and slow. The verifier is interested in only the public key of the target entity. However, the public keys of the intermediate CAs need to be unwillingly found out by the verifier to reach the target entity. This requires multiple inefficient operations to verify the public key of the target entity.

By using the nested certificates, it is possible to verify the subject certificates without employing inefficient signature verification methods based on public key cryptosystems. Therefore, verification of a certificate as a subject certificate of a nested certificate is more efficient than the cryptographic verification. Moreover, it is possible to have a sequence of nested certificates, called *nested certificate path*, to verify a classical certificate at the end. Verification of a certificate via a nested certificate path is also more efficient than the verification of a certificate via classical certificate path.

Verification of a certificate as the subject certificate of a nested certificate is called as *subject certificate verification*. This verification mechanism is a consequence of the nested certificates, but it can be used to verify both classical and nested certificates. Verification of a certificate via a nested certificate path is called as *nested certificate path verification*. This paper mainly deals with the subject certificate verification and nested certificate path verification mechanisms. The subject certificate verification method will be presented first in Section II. In this section, it will also be proven that subject certificate verification has the same confidence as the cryptographic certificate verification. Nested certificate path verification method and similar confidence equivalence proof for this method will be given in Section III. Simulation results for the efficiency improvement of the nested certificate path verification method will be presented in Section IV. The analytical performance analyses can be found in [2,7]. Section V is the conclusions and related work.

II. Subject Certificate Verification Method

The information found out 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 as *subject certificate verification*. Although the subject certificate verification method is a consequence of the nested certificates, it can be used to verify both the nested certificates and classical certificates, since the subject certificates can be of both types.

Subject certificate verification method does not require complex and inefficient public key cryptosystem based operations. Therefore, the subject certificate verification method is faster than the cryptographic certificate verification method.

Having verified the nested certificate, *nc*, in order to verify its subject certificate, *sc*, the verifier follows the following two steps:

1. 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*.

2. 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 in above steps are met and the issuer of *sc* is trusted by the verifier, then the verifier concludes that the *sc* is legitimate. The verifier must trust the issuer of *sc*, since the verification of *sc*, using the above steps, does not mean that the information stored within *sc* is correct. The verifier makes sure about correctness of the information contained in *sc*, if the issuer of sc is trusted. It is very important to point out that, in this way, the subject certificate *sc* becomes verified, with the same confidence, but without employing a signature verification method based on a public key cryptosystem. The phrase “the same confidence” means that the correctness level of the information found out by a subject certificate verification is the same as that of cryptographic certificate verification. Lemma 1 formalizes this conclusion. Moreover, the subject certificate can be another nested certificate. Lemma 2 formalizes the case where the subject certificate is another nested certificate.

Lemma 1: Suppose *A* and *B* are two authorities who are trusted to issue nested or classical certificates. Let *T* be an entity and *V* be the verifier who wants to verify the classical certificate of *T* to find out the correct public key of *T*. Suppose the authority *A* has issued a classical certificate (*cc*) for the entity *T* and the authority *B* has issued a nested certificate (*nc*) for *cc*. Fig. 2a shows the certification relationships. If the *nc* is valid and the verifier *V* trusts both *A* and *B* as nested or classical certificate authorities, then the *cc* can be verified as the subject certificate of the *nc* and this verification has the same confidence as the cryptographic verification of the *cc*.

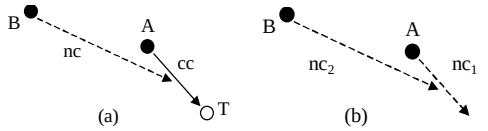

Fig. 2. Certification Relationships in (a) Lemma 1, (b) in Lemma 2

Proof of Lemma 1: By the cryptographic verification of a digital signature over the *cc* using the correct public key of *A*, the verifier *V* can make sure about:

(i) the integrity of the content of *cc* and

(ii) the legitimacy of the signature of *A* over the content of *cc*.

If *A* is trusted by *V*, then the above two results of the verification imply that:

(iii) the information given in *cc* is correct. In other words, the public key of *T* specified in the *cc* is legitimate.

The belief in the integrity of the content of *cc* and the correctness of the signature over it are directly related to the one-way hash functions and the public key cryptosystem algorithms. Here, it will be shown that the above three results can be obtained with the same confidence by the verification of the *cc* as the subject certificate of the *nc*.

By issuing the *nc*, *B* assures the integrity of the *cc* and correctness of the signature of *A* over it. However, this is not a direct assurance that can be verified whenever the *nc* is validated. In the *nc*, *B* gives the correct hash and the signature of *A* over the *cc*. Since the legitimacy of the *nc* and the trustworthiness of *B* are the premises of the theorem, the verifier *V* finds out the correct hash of the *cc* and the correct signature over it. Then, *V* applies the following three steps to verify the *cc* as the subject certificate of the *nc*.

(i) The verifier calculates the hash of the actual *cc*. If the calculated hash is the same as the hash within the *nc*, then *V* concludes that the *cc* has not been maliciously modified after the issuance of the *nc*, because otherwise the calculated hash would differ from the correct hash. In other words, that control verifies the integrity of the *cc*. The integrity control for the cryptographic certificate verification also relies on the comparison of the existing and calculated hash values. Therefore, these two verification schemes have the same confidence for the integrity check.

(ii) The verifier compares the actual signature of *A* over the *cc* with the subject certificate signature of the *nc*. If both of them are the same, then *V* concludes that the *cc* has been signed by *A*. Because, *V* knew the correct signature over the *cc* due to *nc*. Therefore, if the actual *cc* bears the same signature, then *V* can infer that the *cc* contains the correct signature and that signature is issued by *A*. The signature control in this verification scheme has not been done by using public key cryptosystem based operations. However, the necessary cryptographic control had been done by *B* and assurance about it has been given in the *nc* by including the signature over the *cc* as the subject certificate signature. Since

the verifier *V* trusts *B* as an authority and the *nc* is legitimate, *V* makes sure about the correctness of the signature of *A* over the *cc* with the same confidence as if *V* has cryptographically verified the *cc*.

(iii) The verifier *V* made sure about the integrity of the *cc* and correctness of the signature of *A* over it by above two steps. Since *V* trusts *A* as a CA, *V* can make sure about the correctness of the information within the *cc* and consequently finds out the correct public key of *T*. □

Lemma 2: Suppose *A* and *B* are two authorities who are trusted to issue nested certificates. Let *V* be the verifier. Suppose the authority *A* has issued a nested certificate (nc_1) and the authority *B* has issued another nested certificate for nc_1 (nc_2). Certification relationships are shown in Fig. 2b. If the nc_2 is valid and the verifier *V* trusts both *A* and *B* as the nested certificate authorities, then the nc_1 can be verified as the subject certificate of the nc_2 and this verification has the same confidence as the cryptographic verification of the nc_1.

Proof of Lemma 2: Lemma 2 is very similar to Lemma 1. The only difference is that the subject certificate is a classical one in Lemma 1, whereas it is a nested one in Lemma 2. The steps for the cryptographic verification of a nested certificate are the same as the ones for the cryptographic verification of a classical certificate. The only difference is the verified information. By the verification of a classical certificate, the public key of the certificate owner is verified. On the other hand, by the verification of a nested certificate, the correct hash and the legitimate signature over the subject certificate are found. The information within the subject certificate has not been used as a precondition in the proof of Lemma 1. Therefore, Lemma 2 can be proven by following exactly the same arguments as in the proof of Lemma 1.□

III. Nested Certificate Path Verification

In a classical certificate path, each CA can validate the certificates, which have been issued by its immediate successor, since it already knows the public key of its immediate successor. Consequently, each CA can issue nested certificates for all of the certificates that had been issued by its successor. This rule can be applied to a classical certificate path to obtain a structure in which each classical and nested certificate is certified via a nested certificate. From such a structure, it is possible to extract a path of nested certificates and the classical certificate of the target entity at the end. Such a path is called as *k-nested certificate path*, where *k* is the total number of nested certificates on the nested certificate path.

A generic *k*-nested certificate path (the certificates nc_k,nc_{k-1},nc_{k-2} … nc_3,nc_2,nc_1,cc_0) is shown in Fig. 3. 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. Verification of a certificate as a subject certificate is faster than the cryptographic verification of the same certificate. Consequently, nested certificate path verification is more efficient than the classical certificate path verification.

D. Confidence Proof of Nested Certificate Path Verification

In this section, it will be proven that the verification of the classical certificate of the target entity via a nested certificate path has the same confidence as the cryptographic verification of that certificate. Each nested certificate of a nested certificate path is used to verify the next certificate as a subject certificate. The first nested certificate of a nested certificate path is verified cryptographically using the public key of the first entity of the nested certificate path. Lemma 3 formalizes nested certificate path verification assuming that the first nested certificate of the nested certificate path is valid. The verification of the first certificate will be considered in Theorem 1.

Lemma 3: Consider the generic *k*-nested certificate path in Fig. 3. Let A_0 be a CA and A_i, i=1 .. k, be *k* NCAs. Suppose that A_0 has issued a classical certificate (cc_0) for the target entity *T* and A_1 has issued a nested certificate for cc_0 (nc_1). Moreover, suppose that A_i has issued a nested certificate (nc_i) for the nested certificate that A_{i-1} has issued (nc_{i-1}), $\forall i$= 2 .. k. If the nc_k is valid and the verifier *V* trusts the authorities A_i, $\forall i$=0 .. k, then the classical certificate cc_0 can be verified via the *k*-nested certificate path by applying the following steps and this verification has the same confidence as the cryptographic verification of cc_0.

In order to verify cc_0 via the *k*-nested certificate path:

1. First, *V* verifies nc_i as the subject certificate of nc_{i+1}, $\forall i$=k-1 .. 1,

2. finally, *V* verifies cc_0 as the subject certificate of nc_1.

Proof of Lemma 3: The proof is by induction on *k*, the total number of nested certificates on the *k*-nested certificate path. The proof uses Lemma 1 and Lemma 2 of Section II.

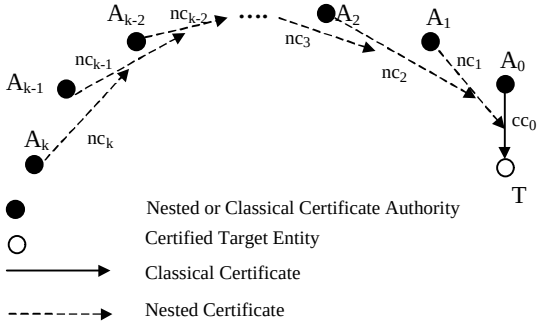

Fig. 3. A generic *k*-nested certificate path

Stage 1 (k = 1): If k = 1, then there is only one nested certificate on the k-nested certificate path. This is the case described in Lemma 1. Therefore by Lemma 1, cc_0 can be verified via nc_1 and this verification has the same confidence as the cryptographic verification of cc_0.

Stage 2 ($k = n$): Assume that if there are n nested certificates on the k-nested certificate path and the nc_n is valid, then the classical certificate cc_0 can be verified via this k-nested certificate path starting with nc_{n-1} and this verification has the same confidence as the cryptographic verification of cc_0.

Stage 3 ($k = n$+1): This is the case where there are n+1 nested certificates on the k-nested certificate path. Since nc_{n+1} is valid, nc_n can be verified as the subject certificate of nc_{n+1} and this verification has the same confidence as the cryptographic verification of nc_n, by Lemma 2. Having verified the nc_n, by the assumption in the stage 2, the classical certificate cc_0 can be verified via the k-nested certificate path and this verification has the same confidence as the cryptographic verification of cc_0. □

Lemma 3 assumes that the first nested certificate of the k-nested certificate path is valid. However, the first nested certificate, nc_k in Fig. 3, of a k-nested certificate path must be verified cryptographically using the public key of the first NCA, A_k in Fig. 3. The complete k-nested certificate path verification is formally given by the next theorem.

Theorem 1: Consider the generic k-nested certificate path in Fig. 3. Let A_0 be a CA and A_i, i=1 .. k, be k NCAs. Suppose that A_0 has issued a classical certificate (cc_0) for the target entity T and A_1 has issued a nested certificate for cc_0 (nc_1). Moreover, suppose that A_i has issued a nested certificate (nc_i) for the nested certificate that A_{i-1} had issued (nc_{i-1}), $\forall i$= 2 .. k. If the verifier V knows the correct public key of A_k and trusts the authorities A_i, $\forall i$=0 .. k, then the classical certificate cc_0 can be verified via the k-nested certificate path by applying the following steps and this verification has the same confidence as the cryptographic verification of cc_0.

In order to verify a k-nested certificate path:

1. Firstly, V verifies the nc_k by employing a public key cryptosystem based signature verification algorithm which uses the public key of A_k,

2. V verifies nc_i as the subject certificate of nc_{i+1}, $\forall i$=k-1 .. 1,

3. finally, V verifies cc_0 as the subject certificate of nc_1.

Proof of Theorem 1: Since the verifier V knows the correct public key of A_k, V can apply cryptographic signature verification algorithm over the nc_k to verify it. After this verification, V can make sure about the validity of nc_k. If the nc_k comes out to be valid, then V can verify cc_0 via the k-nested certificate path and this verification has the same confidence as the cryptographic verification of cc_0, by Lemma 3. □

IV. Simulation Results

Subject certificate verification method does not employ public key cryptosystem operations. Therefore, subject certificate verification is faster than the cryptographic certificate verification. The analytical and simulation based performance evaluation of the subject certificate verification method can be found in [2,7]. Subject certificate path verification method depends on the assumption that the nested certificate of the subject certificate is legitimate. Nevertheless, this nested certificate must also be verified. This reasoning eventually yields a nested certificate path. In this section, performance analysis for the nested certificate path verification method is given.

For the verification of a nested certificate path with n+1 certificates (one classical certificate + n nested certificates), one cryptographic and n subject certificate verifications are performed. On the other hand, n+1 cryptographic certificate verifications must be performed for the verification of a classical certificate path of the same length. That means, n cryptographic certificate verifications are replaced with subject certificate verifications in the case of nested certificate path verification. Since the subject certificate verification method is more efficient than the cryptographic certificate verification, nested certificate path verification is also more efficient than the verification of a classical certificate path of the same length. Moreover, higher relative improvement is expected for the cases of larger n, where n is the number of nested certificates on the path. As a matter of fact, simulation studies show that the efficiency improvement in the nested certificate path verification method is directly related to n.

The simulations are carried out on a Pentium 166 computer using the cryptographic library of the SECUDE toolkit (www.darmstadt.gmd.de/secude). In the simulations, the effect of the number of nested certificates on the nested certificate paths over the relative improvement is examined. The relative improvement measure is the *speed-up factor*, which is the ratio of the verification time of a classical certificate path over the verification time of a nested certificate path of the same length. For the sake of simplicity and uniformity, the same hash algorithms and the same public-key cryptosystems are used for all of the certificates on the paths. Moreover, the number of certificates for both classical and nested certificate paths is the same. Eight sets of simulations are performed; each uses a different pair of public-key cryptosystem (RSA [5] or DSA [6] with different key sizes) and hash algorithm (MD5 [3] or SHA-1 [4]). In each set, the number of nested certificates on the nested certificate paths has been taken in the range of 1 to 8, since these lengths are practical path lengths. Since there is one classical certificate at the end of a nested certificate path, this range corresponds to 2 to 9 total (including nested certificates and the classical certificate) certificates. The results for these simulations are shown in Fig. 4.

As can be seen from Fig. 4, there is a remarkable improvement especially for slower cryptosystems, like DSA-

512, RSA-2048 and RSA-1024. For the cases considered, the speed-up factors are between 1.87 and 8.83. The primary factors that affect the speed-up factor are the number of nested certificates on the path and the public key cryptosystem used for the cryptographic verification. Hash algorithms are much faster than the public key cryptosystem operations, so that the effects of the hash algorithms over the execution times and the speed-up factors are not as significant as the public key cryptosystems. Moreover, hashing is also employed in the subject certificate verification method, whereas public key cryptosystem operations are not. Therefore, the time spent for public key cryptosystem operation is the saving of the subject certificate verification method. That is why the total classical certificate path verification time and consequently the speed-up factor values are larger for the slower cryptosystems, like DSA512, RSA2048 and RSA1024.

As can be seen from Fig. 4, the effect of the MD5 hash algorithm over the speed-up factor is better than the effect of the SHA-1 hash algorithm for the same cryptosystems. The same hash computations are performed for both subject and cryptographic certificate verification methods. Therefore, faster hash algorithm lowers the both verification times. However, public key cryptosystem operations are employed in cryptographic certificate verification also. Therefore, the proportional decrease in the verification time is higher in the subject certificate verification case. Consequently, faster hash algorithm improves the nested certificate path verification time more than the classical certificate path verification time. MD5 is faster than SHA-1. Therefore, the effect of the MD5 algorithm over the nested certificate path verification time and consequently over the speed-up factor is better than the SHA-1 algorithm.

V. Conclusions and Related Work

Nested certificates [2] are used to certify other certificates. Therefore, by using the nested certificates, other certificates can be verified. Verification of a certificate via a nested certificate is called as *subject certificate verification*. In this paper, subject certificate verification method has been described and it has been proven that the confidence of subject certificate verification is the same as the confidence of cryptographic certificate verification.

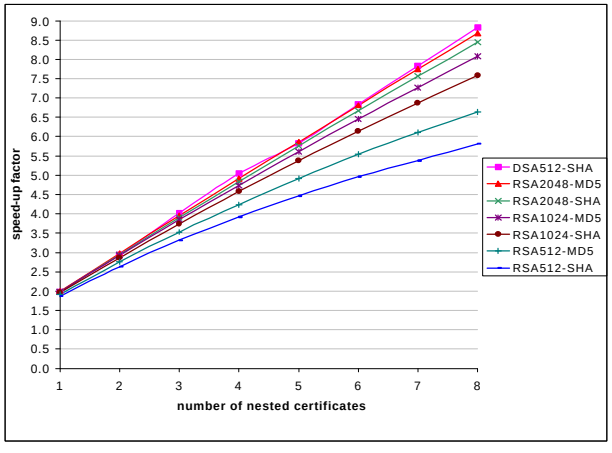

Fig. 4. Simulation results for the change of speed-up factor with respect to the number of nested certificates on the nested certificate paths

Moreover, the nested certificates are useful in the certificate paths. Instead of a sequence of classical certificates, a sequence of nested certificates can be verified to verify a classical certificate. Such a path structure is called as *nested certificate path*. In this paper, nested certificate path verification method has been described. Moreover, it has also been proven that a classical certificate can be verified via a nested certificate path and this verification has the same confidence as the cryptographic verification of the same classical certificate.

Subject certificate verification does not require inefficient public key cryptosystem operations. Therefore, subject certificate verification and, consequently, nested certificate path verification methods are faster than the cryptographic certificate verification and classical certificate path verification methods respectively. Practical simulation results are obtained from the analyses of the nested certificate paths with 1 to 8 nested certificates and by using different cryptosystems and hash functions. Simulation results have shown that, for the cases considered, nested certificate path verification method performs 1.87 to 8.83 times faster than the classical certificate path verification method.

As discussed in [2], the only disadvantage of the nested certificate paths is the *nested certification overhead* for the NCAs. That is, a large number of nested certificates must be issued by the NCAs in order to have nested certificate paths in the global certificate network. The trade-off between the nested certification overhead and the nested certificate path verification improvement has been analyzed in [2] and it has been concluded that the nested certification overhead is acceptable for having efficiently verifiable certificate paths.

References

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

[2] A. Levi, *Design and Performance Evaluation of the Nested Certification Scheme and its Applications in Public Key Infrastructures*, Ph.D. Thesis, Department of Computer Engineering, Bogazici University, Istanbul, Turkey, 1999.

[3] R. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321, 1992.

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

[5] R. Rivest, 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.

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

[7] A. Levi, and M. U. Caglayan, "Analytical performance evaluation of nested certificates," *Performance Evaluation*, vols. 36-37, pp. 213-232, August 1999.