Reducing Certificate Revocation Cost using NPKI

Albert Levi and Çetin Kaya Koç
Oregon State University, Electrical and Computer Engineering Dept., Information Security Lab, Corvallis, Oregon, USA
levi@ece.orst.edu koc@ece.orst.edu

Key words: Digital Certificates, Certificate Revocation, Nested Certificates, PKIs, Digital Signatures

Abstract: Problems with certificate revocation status control limit the deployment of Public Key Infrastructure (PKI). Classical certificate paths require revocation control of all certificates on the path. In this paper, we show how the recently proposed NPKI (Nested certificate based PKI) system reduces the number of revocation status controls to at most two. Our analysis also shows that NPKI is not as vulnerable as classical PKI considering the certificate authority compromise.

1. INTRODUCTION

Certificates are the signed objects that bind the cryptographic public keys of the entities to attributes (name, e-mail address, etc.) or to abilities (file access, fund transfer, etc). They are generated by the digital signature of a *Certificate Authority* (*CA*). The verifiers use the public key of the CA to verify the certificate content. The system that includes the CAs, end users, certificates and certificate management tools is called *Public Key Infrastructure* (*PKI*). Certificates have limited lifespans, but CAs or certificate owners may need to revoke certificates before the expiration time. The reasons of this fact are given below.

- The private key corresponding to the public key in the certificate may be lost or compromised.

- The CA's signature key may be compromised.
- The certification contract may be terminated or the certificate holder's status and abilities described in certificate may change or may be cancelled (as by a person's leaving a job).

Certificate revocation mechanisms must be incorporated into the PKI. The best-known revocation mechanism is the *Certificate Revocation Lists* (*CRLs*). A CRL keeps a signed list of the serial numbers of revoked certificates. Usually, the CA is the signer of the CRL for the certificates that it issued. A good discussion on CRLs can be found in [1].

Another practical revocation mechanism is *Online Certificate Status Protocol* (*OCSP*), which is published as an RFC [2]. OCSP is a simple request/response protocol that requires online servers, so-called OCSP responder, to distribute the certificate status on demand. Each CA must run its own OCSP responder, unless several CAs unite on this issue.

The literature contains other proposed methods of certificate revocation. Micali [3] proposed the use of on-line/off-line signature scheme for a low-cost check for the "freshness" of a particular certificate. Naor and Nissim [4] proposed authenticated data structures to represent CRLs. Kocher [5] proposed Certificate Revocation Trees (CRTs). CRTs are used to compile the revocation information on a single hash tree. Gassko, Gemmell and MacKenzie [6] proposed EFECTS (Easy Fast Efficient Certification System) that combines the best properties of certificates and CRTs. However, their system is best suited for a single CA issuing large numbers of certificates. Rivest [7] proposed an agent based approach that employs on-line "suicide bureaus" to issue "certificates of health" for certificates. A recent certificate of health must be provided to the recipient along with the actual certificate. A brief taxonomy and overview of certificate revocation methods are given by Myers in [8].

CRLs, CRTs or the on-line revocation systems theoretically may become more centralized by having a single revocation authority to process all revocation data on behalf of CAs. Such an approach has the advantage of gathering all revocation information together, but it creates an extra overhead in terms of messaging among the CAs, certificate holders and the revocation authority. Moreover, several CAs must agree to delegate their revocation responsibility to the revocation authority. Therefore, central revocation authority is not suitable for distributed PKIs where CAs of different organizations interact.

Although there may be some exceptional cases where a single CA issues all certificates in a system, the PKI concept inherently employs a topology of several CAs. Therefore, the verifiers should verify a path of certificates in order to learn the public key of an end user. Consequently, they should check

the revocation status of all certificates on the path. To do so the verifier needs to get the revocation information from all CAs on the certificate path. Thus, the difficulty of certificate revocation is multiplied by the amount of CAs (and certificates) on the path. We stress this problem of "distributed" PKIs that has not been addressed in the literature, except in connection with central revocation authorities that are not suitable for distributed PKIs as discussed above.

Nested certificate based PKI (NPKI) [10] is proposed as a model better suited for distributed applications. It allows rapid certificate path verification. In this paper we analyze certificate revocation rules and advantages of NPKI. NPKI facilitates certificate revocation by requiring revocation status check *only* for the first and the last certificate of a certificate path, no matter how many certificates are on the path. A quick introduction to NPKI is given in Section 2. The certificate revocation rules of NPKI are detailed in Section 3. The implications of these rules and the certificate revocation advantage of NPKI are discussed in Section 4. Section 5 is the conclusions.

2. NPKI

NPKI [10] is based on nested certificates. A nested certificate is defined as a certificate for another certificate. A certificate certified in this way is called a subject certificate. A subject certificate can be a classical certificate or another nested certificate. An NPKI is derived from a PKI with all classical certificates that is shown in Figure 1a. Each CA issues one nested certificate for each certificate issued by its children to form NPKI as shown in Figure 1b. A CA must verify a subject certificate before issuing a nested certificate for it. In NPKI, a nested certificate path (e.g. Figure 2a) is produced for each classical certificate path (e.g. Figure 2b) to verify the certificates of the end users.

The PKI-to-NPKI transition does not change the original PKI topology and trust relationships. This can be seen by examining Figures 1 and 2. The same CAs are in control in both PKI (Figure 1a) and NPKI (Figure 1b). The verifier should trust the same CAs in order to verify the classical certificate path of Figure 2a and the nested certificate path of Figure 2b.

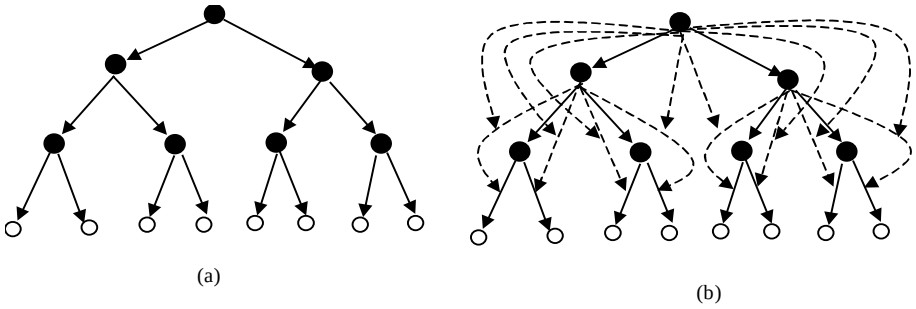
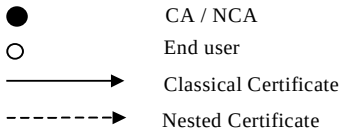


Figure 1. (a) classical PKI, (b) NPKI



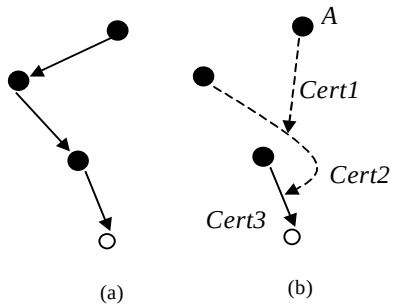


Figure 2. (a) classical certificate path, (b) nested certificate path

The main advantage of NPKI over classical PKI is the improvement in certificate path verification as discussed in [10]. The first nested certificate of a nested certificate path is verified cryptographically. Other certificates, including the last classical one, are verified by hash computations. For example, in Figure 2b *cert1* is verified cryptographically. *cert2* is verified as the subject certificate of *cert1* by only one hash computation. Similarly, *cert3* is verified as the subject certificate of *cert2*. The verifier would need to know only the public key of the first CA (*A* in Figure 2b). The public keys of other CAs are not necessary for path verification.

3. CERTIFICATE REVOCATION RULES OF NPKI

There are some rules about certificate revocation in NPKI. These rules follow the characteristics of NPKI and nested certificates. This section explains certificate revocation rules. The implications of these rules will be discussed in the next section.

Rule 1: Classical certificates are revocable

The classical certificates for the leaf nodes of NPKI may be revoked, as in classical PKIs, if a necessity discussed on Section 1 arises. The guarantees and bindings given in these certificates are invalidated after revocation.

Rule 2: A revoked classical certificate makes its nested certificate path useless

The ultimate aim of a nested certificate path is to verify the classical certificate at the end. Moreover, a nested certificate can exist 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.

Rule 3: Do not start a nested certificate path with a revoked nested certificate, but revoked nested certificates can still be used on paths

If the key of a CA is compromised, then the nested certificates issued by it must be revoked, because these nested certificates must no longer be verified using the public key of the CA. However, this does not mean that these nested certificates contain bogus information. If someone else can prove that these nested certificates were created before the key compromise, they can still be verified. This can be proved by finding another nested certificate issued for the revoked nested certificate before the revocation time. The verifier can verify the revoked nested certificate as the subject certificate of another nested certificate. For example, consider the example in Figure 3. Suppose the CA, *A*, has issued a nested certificate, nc_1, at time t_0. Later at time $t_1>t_0$, another CA, *B*, has issued a nested certificate, nc_2, for nc_1. At time $t_2>t_1$, the public key of *A* is compromised and nc_1 is revoked. After t_2, it is not possible to verify nc_1 using the cryptographic method and the public key of *A*. However, it is still possible to verify nc_1 as the subject certificate of nc_2, which is still valid since *B* had issued nc_2 at time $t_1<t_2$, i.e.,

before the revocation of nc_1. Moreover, *B* had verified nc_1 before issuing nc_2 and guaranteed the legitimacy of the signature over nc_1. The revocation of nc_1 at $t_2>t_1$ does not cause the invalidity of the guarantee given by nc_2 at t_1.

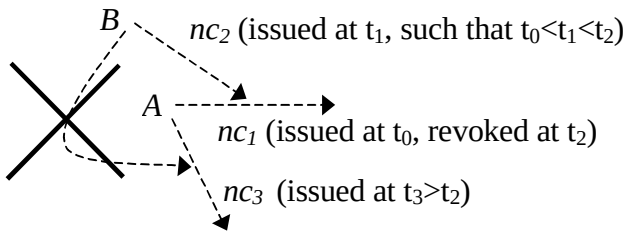


Figure 3. An example case for nested certificate revocation

On the other hand, the counterfeit of *A* can issue some bogus certificates (for example, nc_3 in Figure 3) at $t_3>t_2$, i.e., after the compromise of its key. Since *B* and all other honest CAs are not able to verify nc_3, they will not issue any nested certificates for it. Thus, the bogus certificates remain isolated and cannot take place on nested certificate paths, as long as they are not verified cryptographically as the first certificate of a path.

Rule 4: No cascaded nested certificate revocations

A revoked nested certificate does not cause its subject certificate to be revoked. A nested certificate does not certify a public key or anything regarding a user. A nested certificate certifies only the relationship of the raw content of its subject certificate and the signature over it. The meaning of nested certificate revocation is that the CA of the nested certificate does not guarantee the correctness of the signature over the subject certificate anymore. However, the signature over the subject certificate can still be verified cryptographically using its issuer's public key. Therefore, nested certificate revocation is not a recursive process towards the end users.

4. DISCUSSION

The above rules imply that the verifier must check the revocation status of two certificates on a nested certificate path regardless of the path length. One of them is the first nested certificate, which is to be verified cryptographically. This certificate must be checked in order not to start the verification process with a bogus certificate (rule 3). Second certificate for which the revocation status must be checked is the last certificate of the

nested certificate path, because it is a classical certificate and the revoked classical certificates cannot be used (rule 1). Other nested certificates on the path need not be checked for revocation, because even if an intermediate nested certificate is revoked this does not cause other certificates to be revoked (rule 4) and it can be used on the path (rule 3).

However, in a certificate path of a classical PKI, all certificates must be checked against revocation. Since all these certificates are from different CAs, different CRLs or OCSP responder contacts would be necessary for the revocation checks. Since there are only two certificate revocation controls in NPKI, the cost of certificate revocation relatively decreases for the paths longer than two certificates as compared to classical PKI.

The revocation status of the first nested certificate of a nested certificate path must be checked since it might have been revoked due to a CA key compromise as discussed in rule 3. This revocation control can be waived if the verifier can make sure about the legitimacy and validity of the public keys that it uses to start the verification process. This would be possible by keeping this CA key information in a local Personal Security Environment (PSE) and by periodically checking the validity of these keys. Similar approaches are proposed by PGP [11] and ICE-TEL [9] systems. However, the revocation status of the classical certificate at the end of a nested certificate path must always be checked.

One can argue that the CA compromise might go undetermined for a long time and during this period some bogus nested certificates can be disseminated. This is still not a big problem and does not require a mass revocation of innocent certificates. Once the breach is detected, it is sufficient to revoke the certificates issued by the compromised CA after the compromise, and the nested certificates issued on them recursively[†]. One may also argue that the counterfeit may change the timestamps in the certificates as if they are issued earlier. This is not correct, because if the counterfeit does so, other CAs realize that something is going wrong and decline to issue nested certificates for the certificates issued by it. Thus, bogus certificates remain isolated.

Above discussion and the rules 3 and 4 also yield that CA compromise in NPKI is not as severe as in classical PKI. The main reason behind this fact is that each CA controls its children by the nested certification process embedded in NPKI. There is no such control in a classical PKI. Once a classical certificate is issued, the issuer can no longer control the activities of the certificate holder.

[†] If this argument is the concern of the system, the verifier should check the revocation status of the first certificate of the path even if he/she makes sure about the validity of the public key of the corresponding CA, because this argument brings out a reason other than CA key compromise to qualify a nested certificate revoked.

Revoked certificates can be kept in Certificate Revocation Lists (CRLs) or handled by other methods cited in Section 1. Each CA manages its own revoked certificates. There may also be nested certificates that are not revoked but are useless (rule 2). 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.

5. CONCLUSIONS

Nested certificate based PKI (NPKI) has been recently proposed as an efficient, dynamic and trust-preserving PKI scheme [10]. In this paper we analyzed the revocation characteristics of nested certificates and NPKI. We concluded that it is sufficient to check the revocation status of at most 2 certificates on a nested certificate path, the first and the last certificates, regardless of the number of certificates on the path. The rule for "classic" PKI is to check the revocation status of all certificates on the path, giving NPKI an obvious advantage.

Our analysis also indicates that NPKI CAs are less vulnerable to being compromised than PKI CAs, since their activities are monitored via nested certification.

NPKI does not add any extra burden to facilitate certificate revocation and to make their CAs less vulnerable. These characteristics are the consequences of the nested certification scheme embedded in NPKI.

ACKNOWLEDGMENTS

This work has been supported by rTrust Technologies.

REFERENCES

1. Adams, C., and S. Llyod, *Understanding Public Key Infrastructures*, New Riders Publishing, 1999
2. Myers, M., R. Ankney, A. Malpani, S. Galperin, and C. Adams, X.509 Internet Public Key Infrastructure On-line Certificate Status Protocol – OCSP, RFC 2560, June 1999.
3. Micali, S., Efficient Certificate Revocation, MIT Laboratory for Computer Science, Technical Memo 542b, March 1996.
4. Naor, M., and K. Nissim, "Certificate Revocation and Certificate Update," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 561 – 570, April 2000.

5. Kocher, P., “On Certificate Revocation and Validation,” Proceedings of Financial Cryptography 98, LNCS 1465, Springer-Verlag, pp. 172-177, Anguilla, BWI, February 1998.
6. Gassko, I., P. S. Gemmell, and P. MacKenzie, “Efficient and Fresh Certification,” Proceedings of Public Key Cryptography (PKC) 2000, LNCS 1751, Springer-Verlag, pp. 342-353, Melbourne, Australia, January 2000.
7. Rivest, R., “Can We Eliminate Certificate Revocation Lists?,” Proceedings of Financial Cryptography 98, LNCS 1465, Springer-Verlag, pp. 178-183, Anguilla, BWI, February 1998.
8. Myers, M., “Revocation: Options and Challenges,” Proceedings of Financial Cryptography 98, LNCS 1465, Springer-Verlag, pp. 165-171, Anguilla, BWI, February 1998.
9. Chadwick, D. W., A. J. Young, and 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.
10. Levi, A., and M. U. Caglayan, “An Efficient, Dynamic and Trust Preserving Public Key Infrastructure", Proceedings of 2000 IEEE Symposium on Security and Privacy, pp. 203 - 214, Oakland, CA, USA, May 2000.
11. Zimmermann, P., PGP User's Guide, available with free PGP software from http://www.pgpi.com.