A NOVEL APPROACH TO CERTIFICATE REVOCATION MANAGEMENT

Ravi Mukkamala and Sushil Jajodia

Abstract: With the ever-increasing growth in electronic messaging and electronic commerce, the need for an infrastructure to provide confidentiality, security, and confidence for such exchanges to take place is quite evident [2]. Here, public keys and certificates are issued to users for authorization purposes. One of the primary concerns in these systems is the handling of certificate revocation prior to the expiration date. In this paper, we propose a new approach for managing certificate revocation. All existing schemes require that the information about revoked certificates be sent only periodically to the directories used for verification. This gives rise to the problem of obsolescence. To overcome this problem, we have introduced a new layer in the traditional architecture. Using a preliminary analysis, we show the impact of the new scheme on the up-to-datedness, robustness, load distribution, and response time of the system. Similarly, we show the additional costs incurred in terms of communication cost, processing cost, and hardware costs.

Keywords: Certification revocation list (CRL), public-key infrastructure (PKI)

1. INTRODUCTION

With the ever-increasing growth in electronic messaging and electronic commerce, the need for an infrastructure to provide confidentiality, security, and confidence for such exchanges to take place is quite evident [2]. Public-Key Infrastructure (PKI) is one such mechanism [2, 5, 8, 10]. Here, public keys are issued to users. In addition, certificates are issued to certify that a given public key does indeed belong to a user.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35587-0_24

M. S. Olivier et al. (eds.), *Database and Application Security XV*

A certificate attests to the validity of a public key of an individual or other entity and is normally issued by a *Certification Authority* (CA). The purpose of a digital certificate is to allow verification of the claim that a specific public key does in fact belong to a particular entity or individual. Each certificate contains information such as the certificate version number, serial number, name of the issuer (i.e., the CA) and the subject (i.e., the person or entity to whom the certificate relates), a validity period (i.e., not valid before, and not valid after, specific dates/times), a public key, the CA's signature, and the signature algorithm used [2].

Once a Certification Authority (CA) issues a certificate, it is assumed that the certificate will be in use for its entire validity period. However, various circumstances such as compromise or suspected compromise of the private key may cause a certificate to become invalid or untrustworthy prior to its expiration date. Under such circumstances, the CA needs to revoke the certificate [16,18].

Several schemes have been proposed for maintaining and propagating the certificate revocation information. A certificate revocation list (CRL) is one such mechanism [3, 4, 6, 9, 11, 16]. Each CRL is a digitally certified list that identifies revoked certificates with its serial number and the date of revocation. This list could be significantly large depending on the number of certificates revoked by a CA prior to their expiration.

When a certificate is used, a check should be made (say, by a service-provider) on the certificate's signature and validity. The recipient of a certificate from an unknown or untrusted source can check the validity and trustworthiness of the certificate and the public key that it contains by verifying the signature contained in the certificate and checking if the certificate has been revoked or not. For this purpose, the service-provider seeks the services of a "directory" entity. CAs provide the information to the directories to offer such a verification service. When CRLs are used, the directory provides the service-provider with the latest CRL. The service-provider, in turn, can check whether or not the given certificate serial number is in the received CRL. The key question that the service-provider may have about the verification process is: "How much can I rely on the information provided by the directory service?" We refer to this as an issue of confidence that the service-provider is offered by the rest of the system. The answer to this question depends on the "currency" or "up-to-datedness" of the revocation information stored at a directory.

In this paper, we suggest a novel scheme to improve the confidence in the certification verification checking system. This is achieved by introducing a new layer in the multi-tier architecture. The entities at this layer are referred to as CADs (or CA-directories). The proposed architecture also improves the robustness of the certification verification system. In addition, it improves the system performance by transferring some of the CA's load on to its CADs (as explained later). This is shown using a preliminary analysis.

The paper is organized as follows. Section 2 briefly summarizes existing schemes to assemble and propagate revocation information. Section 3 describes the proposed architecture for disseminating the revocation information. Section 4 analyzes the proposed architecture in terms of its currency of information, robustness, and cost. Finally, Section 5 summarizes the paper and discusses possible future extensions to this work.

2. CURRENT SCHEMES FOR CERTIFICATE REVOCATION AND CHECKING

Several schemes have been proposed in the last few years for certificate revocation and dissemination of revocation information. The traditional method is the certification revocation lists (CRL), which are time-stamped, digitally signed lists containing the certificate serial numbers and the revocation dates of all revoked certificates within a CA. A CA typically publishes them periodically and sends them to the directories. The CRL scheme is included as part of the X.509 standard. CRLs are criticized for not providing adequate service and being too costly [14, 17]. However, at present CRLs are in wide use.

The Delta CRL scheme is a mechanism that addresses the size of the CRLs [7]. A Delta CRL is a digitally signed list of incremental changes that took place since the last posting of a full CRL. Delta CRLs are generally significantly smaller than full CRLs, and hence can be posted more frequently. Delta CRLs reduce the network load but may increase the processing load at a directory when it needs to service a verification request from a service-provider.

One of the major problems with CRLs having the same expiration date is that all service-providers may find that their CRLs have expired within a short time. This implies that a directory (or a repository) may suddenly be overloaded with such requests for CRLs. This is rectified by over-issued CRL policy where a CA issues CRLs with different expiration times. This means that depending on the time that a service-provider gets a CRL from its directory, the expiration date may be different. This distributes the request load at the directories. This is referred to as an over-issued CRL method [6, 7].

An alternate scheme to improve performance is the indirect CRL method [1]. Here, a different entity besides a CA can issue a CRL. In addition, a CRL can contain information from multiple CAs. This reduces the number of CRLs that need to be transmitted over the network. It is, however, assumed that the entity issuing the CRLs is trusted.

Certificate Revocation Status (CRS) directory was an early suggestion for reducing the large communication overhead in disseminating the CRLs [14]. In this scheme, a CA generates a 100-bit value for each unexpired certificate. The value would be different for a valid one and a revoked one. The CA periodically updates the directory with the 100-bit values as well as a list of all certificate serial numbers that have not expired. This scheme drastically reduces the communication load.

Other variations of the basic CRL scheme are the Certificate Revocation Trees (CRT) [16], and the On-line Certificate Status protocol (OCSP) [15].

In the rest of the paper, for simplicity, we assume the basic CRL protocol. The discussion and the analysis, however, are independent of the actual protocol and hence applicable for other schemes also.

3. PROPOSED SCHEME FOR CERTIFICATE REVOCATION AND CHECKING

One of the deficiencies of the current schemes (both CRL and CRS) [6, 14] is that the certification authorities only send periodic updates of revocation to the directories. In addition, the service-providers can only contact a directory (and no one else) for revocation status of a certificate. Thus, the up-to-datedness of the service offered by a directory to service-providers depends on the periodicity with which a CA updates the revocation information at the directory. Since there may be hundreds of CAs and thousands of directories [2] in a PKI system, the cost of keeping directories in sync with the CAs would be astronomical. On the other hand, keeping them out of sync reduces the confidence that the service-providers have about the PKI system. We propose an architecture that has a means to improve the service-providers' confidence.

Figure 1 illustrates the five-level PKI architecture. At level 1 are the users who submit requests for services to the service-providers. Along with a request for service, they also provide a public key and a certificate. The service-providers are at level 2. When a service-provider receives a request accompanied by a certificate from a user, it locally checks for the validity of the digital signature. Occasionally, it may contact a directory (at level 3) to check whether or not the certificate presented has been revoked prior to its expiration date. Since a directory itself is only updated with the revocation information periodically (and not as and when the revocation takes place), the service-provider has the option of contacting a CA-directory (CAD) at level 4 to verify the latest status of a certificate. In other words, a service provider has three choices when a certificate is presented to it---use local information, check with a directory, or check with a CAD.

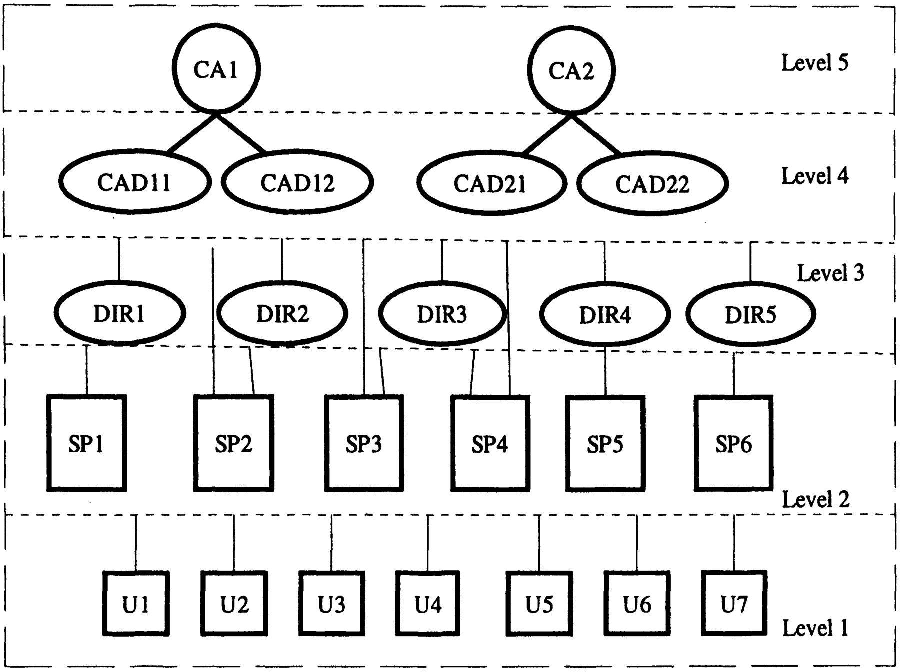

Figure 1. Proposed Architecture for Certificate Revocation and Checking

At level 3, we have directories that are untrusted or less trusted entities. Each directory contains information about the status of certificates issued by all the CAs in the system. In the case of a CRL-based system, a directory would have one CRL per CA. Whenever a service-provider sends a revocation-check request, the directory simply returns the CRL corresponding to the CA that issued the certificate. In the case of CRS, however, the directory sends a certified status information related to that certificate.

At level 4, we have the CADs or CA-directories. This is the additional level introduced in this paper to improve the up-to-datedness of revocation information. A given CAD (such as CAD11 in Figure 1) is associated with a single CA (such as CA1 in Figure 1). Since a CA does not get involved in the verification of certificate status [6], a CAD acts as a proxy for a CA in this task. For this reason, it is a *trusted entity*. It also should be noted that unlike the directories that simply act as intermediaries in delivering the CA (or CAD) supplied CRL to the service providers, a CAD needs to construct the CRLs. This calls for a *trusted CAD* since an untrusted one can generate its own CRLs irrespective of the input from its CA. A CA treats its CADs as its own replicas and keeps them informed of the certificate revocations (and optionally, certificate issues) as and when they occur. Whether or not such an update between the CA and its CADs is done atomically is an implementation issue. But our architecture does not mandate such an atomic update. A CAD keeps information about its own CA. This model can, however, be extended so that it can also act like a directory for other CAs. In other words, a CAD may contain most up-to-date information about its own CA's certificates but contain only periodically updated information about certificates issued by other CAs. This is an implementation and performance issue. For simplicity, we assume that a CAD only contains information about its CA. A CAD is responsible for periodically sending the revocation information about its own CA to all the directories. (In traditional architectures, this was a CA's responsibility.)

Finally, at level 5 are the CAs. These are trusted entities that issue and revoke certificates. Users directly contact a CA for the issue and revocation of certificates. CAs update the information immediately at their CADs. Table 1 summarizes the tasks of the entities at each level of this architecture.

4. COST-BENEFIT ANALYSIS

In this section, we analyze the proposed architecture in terms of the cost incurred and the benefits reaped. The primary cost is the communication cost among the levels and the processing cost at each entity. In addition, cost of additional hardware is also to be considered. The benefits are measured in terms of up-to-datedness of the information, robustness, load distribution, and response-time. In the following discussion, we model the system under a steady state. Thus, we do not model the system behaviour under transient conditions.

C1. Communication cost: Let us now consider the cost of adding an additional level, the CAD level, into the conventional architecture [2, 6, 14]. The interaction between the CA and CADs is an addition to the original architecture.

Every time there is either a new certificate issued or a certificate is revoked, CA would send a message to its CADs. If the rate of certificate issue per CA is λ_{CI}, the length of a certificate is L_{CI}, and the number of CADs per CA is N_{cad}, then the rate of additional messages transmitted (per CA) due to certificate issue is λ_{CI}* N_{cad}. Similarly, the rate additional bytes transmitted (per CA) is given by λ_{CI}* N_{cad}* L_{CI}.

Every time there is a certificate revocation, CA would send a message to its CADs. If the rate of certificate revocation per CA is λ_{CR}, and the length of certificate revocation information is L_{CR}, the rate of additional messages transmitted due to certificate revocation is λ_{CR}* N_{cad}. Similarly, the rate of additional bytes transmitted is given by λ_{CR}* N_{cad}* L_{CR}.

In summary, the rate of additional messages transmitted is $N_{cad}(\lambda_{CI}+\lambda_{CR})$ and the rate of additional bytes transmitted is $N_{cad}(\lambda_{CI}*L_{CI} +\lambda_{CR}*L_{CR})$. This does not include any messages that may be involved if a CA wants to carry out the updates in an atomic fashion. Since this depends on the specific implementation, we have not included it here.

C2. Processing cost. First, due to the transfer of the certificate status distribution functionality from a CA to its CADs, the processing load at a CA is significantly reduced. This improves the performance that it offers to users for certificate issues and certificate revocations. Additional processing, however, is introduced for updating the CADs as and when a CA issues or revokes certificates. This cost is reflected more in additional communication rather than in processing. Depending on the implementation, there could be a cost of synchronization between a CA and its CADs during such updates.

The processing at a CAD is, of course, an additional cost compared to the traditional architectures. A CAD participates in updating the status information with a CA---it updates the CRLs or CRSs, and distributes them to the directories periodically. In addition, it answers occasional queries from a service-provider regarding the status of a certificate. In summary, some of the processing cost is transferred from CAs and directories to the CADs. Since we assume them to run on independent processors, it would not overload any existing system. The actual impact depends on the implementation.

C3. Hardware and set-up cost. The CADs need additional hardware to run. Besides this cost, there would be no additional costs under this category.

B1. Up-to-date certificate information. In the traditional model, the CA sends the status information of its certificates to the directories periodically. So if the period of status update is T_{SU}, then the information at the directories may be out-of-date by as much as T_{SU}. Suppose the period is 1 week. Then a directory may be unaware of the revocation of a certificate for as long as a week. The cost of providing stale revocation information is borne by the service-providers. When the rate of certificate revocation is λ_{CR}, one may expect an average of $\lambda_{CR}*T_{SU}$ certificates to be revoked in the interval between two updates from CA.

The cost of obsolescence in revocation information can only be measured by its effect on a service-provider. Suppose the service provider assumes that the cost of obsolete information (i.e., providing service to a user whose certificate has been revoked) is C_{ex}. If a service-provider receives service requests at the rate of λ_{sr}, then a fraction of them may be submitting unexpired but revoked certificates. If we assume that a user with an unexpired but revoked certificate continues to behave as if it was never revoked, and if we assume that a service-provider does not keep track of the so called bad credit, then on the average $\lambda_{sr}* P_{rev}$ requests may be corresponding to revoked certificates. This may represent a loss of $C_{ex}*\lambda_{sr}* P_{rev}$ for the service-provider. To reduce this loss, a service-provider contacts the directory service to check the status of a certificate. However, there is a cost C_{cr} to confirm the status of a certificate. Hence, a service provider checks with a directory only with a probability P_{dir}. However, even with this cost, it cannot be completely confident about the status of a certificate.

In the proposed scheme, the service-provider has an option. If it wants to confirm the status of a certificate with 100% confidence, then it could send a verification request to the CAD. Since there are only a few CADs for each CA, the delay at a CAD is likely to be higher than the one at a directory. This could be considered as a cost. In addition, there will be a confirmation cost (similar to C_{cr} and probably higher). As a trade-off, a service- provider could send a fraction of requests to a directory and another fraction to a CAD. Clearly, a high-value request (e.g., high-value purchase) is to be sent to CAD and a medium-value request to a directory. In this way, the service-provider could reduce its losses due to revoked certificates. Of course, if a service-provider needs 100% confidence about a certificate prior to providing a service, then all status confirmations should be sent to CADs. If it prefers to have larger throughput than confidence in the status, then it may be satisfied with its own local validation. In general, it may divide its requests among the local validation, directory, and CAD.

In order to illustrate the effect of introducing CADs on the up-to-datedness of revoked information, we have simulated a system with thousands of generated certificates. Assuming the probability that a certificate is revoked during its lifetime is given, for a given lifetime (T_{LC}) of a certificate, we randomly determine the characteristics of a certificate---the generation time, whether or not it is revoked, and if so when it is revoked. We then vary the status update time (T_{SU}) as a fraction of T_{LC} and measure the impact of CADs. Figure 2 illustrates the results. We measure the impact of introducing CADs on the up-to-datedness in terms of the ratio of the number of unexpired certificates recoded in a CRL to the actual number of unexpired certificates. For example, if 1000 actual unexpired revoked certificates exist of which only 800 were included in the last CRL (for example), the measure of up-to-datedness is 800/1000 or 80%. This metric in some sense indicates the degree of up-to-datedness gained due to the introduction of CADs in the architecture. When Metric 1 is 80%, it indicates that 20% of the queries to a directory would be receiving incorrect answers. I.e., 20% of the time, even a revoked certificate would be reported as unrevoked by the directory. On the other hand, when a service-provider sends its request to a CAD, there would be no such errors.

While the basic definition of the metric is straightforward, the actual measuring process is not unique. One of the key questions that we encounter is the time at which this ratio be measured. In addition, suppose the measure is 70% during 100 units of time and 30% during 30 units, then should we take the time-weighted average as a representative value?

Since the answer was non-trivial, we decided to measure the up-to-datedness by two means. First, we measure it as a time-weighted average. This is referred to as Metric 1. To evaluate this metric, we identified three types of events in the system (i) CRL generation (ii) Certificate revocation, and (iii) Expiry of revoked certificate. At each of theses events, we measure the ratio defined above and then weight it with the time this ratio persists. The weighted average of these measures is represented as Metric 1. As an alternate, we measured the ratio at each instant of certificate revocation (i.e., event 2 of Metric 1), and averaged the ratio over all such events. This is represented as Metric 2.

In Figure 2, it is assumed that a certificate may be revoked at any given time during its lifetime with a probability of 0.1. In other words, 10% of unexpired certificates may be revoked [2, 14]. From the graph it is clear that as the ratio of TSU to TLC increases, the benefit of the proposed architecture increases. In other words, as the interval between CRL updates to directories increases, the up-to-datedness of CRL in directories in the traditional architecture decreases. This is, of course, avoided in the proposed architecture due to the presence of CADs. It may be noticed that the complex measure of Metric 1 as well as Metric 2 resulted in very close results. This may not be too surprising given that the transactions had random start time and revocation times (if revoked). The randomness in the transactions basically eliminated the time factor that we took into account in Metric 1.

Not surprisingly, the rate of arrival of transactions had negligible effect on metric 1 and metric 2. For example, when the rate of revoked transactions was doubled, then metric 1 changed from 76.8% to 77.1%. When it was tripled, it was still 77.1%. Similar behavior was observed of Metric 2. We are still investigating other properties of these metrics.

B2. Robustness: In the traditional architecture, there is a single CA responsible for the issue and revocation of certificates for a given group of users. In the proposed architecture, we have one or more CADs acting as proxies for the CA. Hence, temporary failure of a CA does not result in loss of revocation-checking service offered to the service-providers. If we assume that each service-provider first approaches its own CAD for revocation-checking, and in case of its unavailability seeks the services of other CADs, and ultimately, when all CADs are unavailable, seeks the CA itself, then the robustness offered for the revocation-checking service is given by the probability that at least one of CADs or CA is available. This may be expressed as $1-[(1-R_{CA})\Pi_{i=1,Ncad}(1-R_{CADi})]$, where R_{CA} is the probability that a CA is available, R_{CADi} is the probability that the i^{th} CAD is available, and N_{cad} is the number of CADs per CA. Given the mean time between failures (MTBF) and the mean time to repair (MTTR) of a CA or CAD, the probability that a component is available can be computed as MTBF/(MTBF+MTTR). A CA or CAD may be unavailable either due to planned events such as backup or other maintenance functions, or due to unplanned events such as power failure, system crash, or communication failure.

Figure 3 illustrates the benefit of adding the new layer to the robustness of the system. The ratio MTBF/(MTBF+MTTR) represents the fraction of time that a CA or CAD is available. The benefit of CADs is clearly evident from this graph---as the number of CADs per CA is increased, the robustness is also increased. Of course, the incremental benefit beyond 2 CADs is not significant.

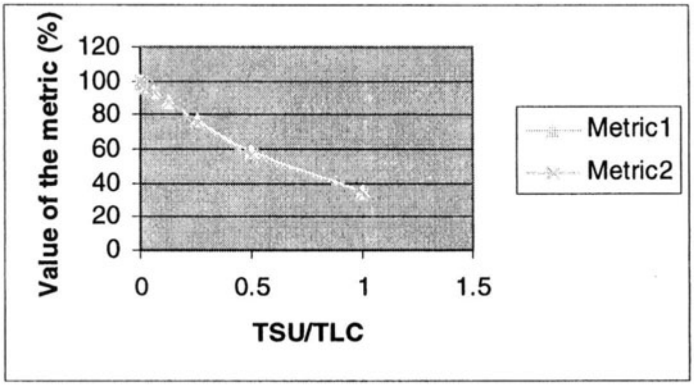

Figure 2. Impact on Up-to-datedness of revocation information

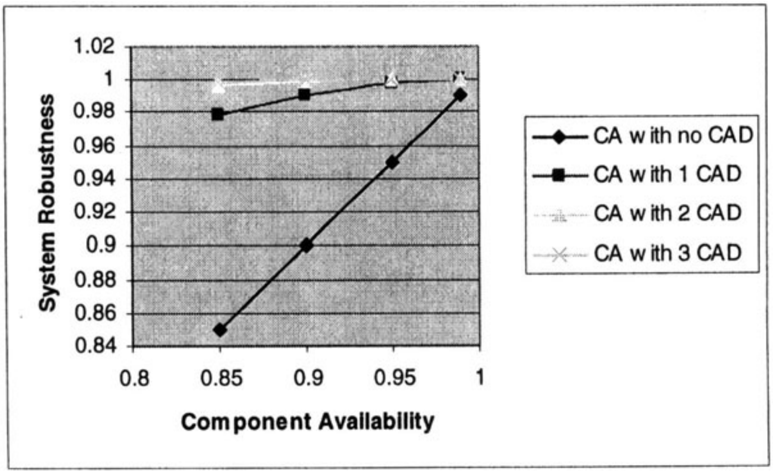

Figure 3. Robustness offered by the proposed architecture

B3. Load distribution: In the traditional architecture, in addition to the responsibility of issuing and revoking certificates, a CA is also responsible for distributing this information to the directories. In fact, this was of primary concern when the number of users and directories under a CA is large [6, 7]. In the proposed architectures, the latter responsibility of distributing the information to directories is delegated to the CADs. Typically, a CAD acts as a primary distributor of information for a specified set of directories. Only in the event that it is not available, either another CAD or the CA itself take up this responsibility. Accordingly, if the rate of periodic certificate status information to each of the N_{dir} directories is λ_{stat} and processing time to prepare and send a message is t_{stat}, then we have effectively eliminated a processing load of $N_{dir}*t_{stat}*\lambda_{stat}$ (per unit time) at the CA and distributed the same among the N_{cad}. Instead, the CA needs to inform each of its CADs whenever it issues or revokes a certificate. If processing time to send a revoke or issues information to a CAD is t_{ca}, the rate of revocation is λ_{CR} and the rate of certificate issue is λ_{CI}, then the additional rate of processing at a CA due to CADs is $t_{ca}*N_{dir}*(\lambda_{CI} + \lambda_{CR})$. This, of course, is a simplified model assuming that the time is proportional to the number of CADs. In general, it may be a more complex function. Thus, the net reduction in the rate of processing load is $(N_{dir}*t_{stat}*\lambda_{stat} - t_{ca}*Ndir*(\lambda_{CI} + \lambda_{CR}))$. This will be reflected in a change in the response time to the user's certificate issue and revocation requests at the CA.

B4. Response time. As mentioned earlier, the transfer of load from CA to CADs improves the response time offered by CAs for certificate issue and revocation requests. Similarly, since some status requests are now routed to CADs, the response at the directories will also be improved. Since CADs share the status query requests among themselves, and only a fraction of queries are received from the service-provider even they can provide a good response. In this paper, we did not quantify this benefit.

5. CONCLUSION

In this paper, we have proposed a new approach for managing certificate revocation in an electronic authorization scheme such as the public-key infrastructure. In all the existing schemes, the information about revoked certificates is only periodically sent to the directories that are used for verification. This gives rise to the problem of obsolescence. To overcome this problem, we have introduced a new layer in the traditional architecture. We refer to it as CAD or CA-directories layer. We then showed the impact of the new scheme on the up-to-datedness, robustness, load distribution, and response time of the system. The additional costs are measured in terms of communication cost, processing cost, and hardware costs. While the actual degrees of benefits depend on factors such as the frequency with which CRL updates are sent to directories and the percentage time a node is available, the benefits are visible over a wide range of these factors.

In the future, we propose to do a detailed simulation of the scheme and measure its other characteristics. In addition, we plan to study schemes under which more and more responsibilities are transferred from CA to CADs thereby improving CA's performance.

REFERENCES

[1] Adams, C., and S. Lloyd, Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations, Macmillan Publishing, 1999.

[2] S. Berkovits, S. Chokhani, J. A. Furlong, J. A. Geiter, and J. C. Guild, "Public Key Infrastructure Study: Final Report," MITRE, McLean, Virginia, 1994.

[3] CertCo, Inc. White Paper by Richard C. Ankney: Certificate Revocation Mechanisms. Retrieved March 01, 2001 from the World Wide Web: http://www.certco.com./

[4] CertCo, Inc. White Paper by Richard Salz: The Transaction Instant: A Question of Validity. Retrieved March 01, 2001 from the World Wide Web: http://www.certco.com/b2b/OCSP_Salz.pdf

Level	Entity	Type	Functions
5	CA	Trusted	Responsible for issuing and revoking certificates. Inform CA-DIR about issues and revocations.
4	CAD	Trusted	Contains up-to-date information on the status of issued certificates. Responsible for verifying certificate status and periodically informing the directories about revocations.
3	Directory	Untrusted	Contains information that is periodically updated. Responsible for verifying certificate status.
2	Service-provider	Untrusted	Provides services after verifying the validity of users' certificates.
1	User	Untrusted	After obtaining certificates from CA, requests services from service-providers.

Table 1. Architectural entities and their functionality

[5] Chokhani, S., "Toward a National Public Key Infrastructure," IEEE Communications Magazine, September 1994, Vol. 32, Issue: 9, pp. 70-74.

[6] Cooper, D.A., "A model for certificate revocation," Proceedings of the 15th Annual Computer Security Applications Conference, pp. 256-264, December 1999.

[7] Cooper, D.A., "A more efficient use of Delta-CRLs," Proceedings of the 2000 IEEE Symposium on Security and Privacy, pp. 190-202, May 2000.

[8] Ellison, C. and B. Schneier, "Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure." Computer Security Journal, v 16, n 1, 2000, pp. 1-7. [Magazine, selected stories on-line]. Retrieved March 01, 2001 from http://www.counterpane.com/pki-risks.html.

[9] Fox, B., and B. LaMacchia, "Certificate revocation: Mechanics and Meaning," URL: www.farcaster.com/papers/fc98/.

[10] Fratto, M., "Certificate Revocation: When not to trust," Network Computing, June 26, 2000 (URL: http://www.networkcomputing.com/1112/1112ws1.html)

[11] Housley, R., W. Ford,et al., "X.509 Internet Public Key Infrastructure Certificate and CRL Profile." The Internet Society,1999. Retrieved February 14, 2001 from ftp://ftp.isi.edu/in-notes/rfc2459.txt.

[12] McDaniel, P., "Windowed Certificate Revocation," Technical Report CSE-TR-413-99, Department of Electrical Engineering and Computer Science, University of Michigan, 1999 (URL: http://www.eecs.umich.edu/techreports/cse/1999/CSE-TR-413-99.pdf)

[14] Micali, S., "Efficient certificate revocation," Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, 1996.

[15] Myers, M., R. Ankney, A. Malpani, S. Galperin, and C. Adams, X.509 Internet Public Key Infrastructure: On-line Certificate Status Protocol, IETF RFC 2560, June 1999.

[16] Naor, M., and K. Nissim, "Certificate revocation and certificate update," Proceedings of the 7th USENIX Security Symposium, 1998.

[17] Rivest, R.L., "Can we eliminate certificate revocation lists?" Proc. Financial Cryptogrpahy 1998, Springer-Verlag, Feb. 1998.

[18] Stubblebine, S., "Recent Secure Authentication: Enforcing Revocation in Distributed Systems," IEEE Symposium on Research in Security and Privacy, Oakland, May, 1995, pp. 224-234.