Certificate Revocation Lists or Online Mechanisms[1]

Vipul Goyal

Department of Computer Science & Engineering
Institute of Technology
Banaras Hindu University
Varanasi, India

Abstract. With more and more acceptance of Digital Certificates and Public Key Infrastructures (PKI), the mechanisms to revoke a certificate in a PKI have recently received increasing attention. The revocation mechanisms are commonly classified into Certificate Revocation Lists (CRLs), trusted dictionaries and online mechanisms. The designer of a PKI should select an appropriate revocation method suiting his requirements. This turns out to be a sufficiently confusing task as different revocation solutions are good in different type of environments. We ask the question "How do we decide which revocation solution to use amongst the various categories of solutions?" We first conduct a survey of the existing certificate revocation techniques and then analyze and compare the various classes of revocation methods for their advantages and disadvantages. This analysis can greatly help the PKI designer to select the right revocation solution.

1 Introduction

A certificate is a digitally signed statement binding the key holder's (principal's) name to a public key and various other attributes. The signer (or the issuer) is commonly called a certificate authority (CA). Certificates act as a mean to provide trusted information about the CA's declaration w. r. t. the principal. The declaration may be of the form-

"We, the Certificate Authority declare that we know Alice. The public key of Alice is ..."

"We further declare that we trust Alice for ..." (optional part)

Certificates are tamper evident (modifying the data makes the signature invalid), unforgeable (only the holder of the secret, signing key can produce the signature). Certificates are the building blocks of a Public Key Infrastructure (PKI). PKI is defined to be "The set of hardware, software people, policies and procedures needed

[1] Throughout the paper, we use examples involving Alice and Bob, where Alice is assumed to be the sender and the subject of the certificate and Bob is assumed to be the acceptor (or the verifier) of the certificate.

Goyal V. (2004).
Certificate Revocation Lists or Online Mechanisms.
In *Proceedings of the 2nd International Workshop on Security in Information Systems*, pages 261-268
DOI: 10.5220/0002679502610268

to create, manage, store, distribute, and revoke public key certificates based on public key cryptography" in the IETF PKIX Roadmap [1].

When a certificate is issued, the CA (issuer) declares the period of time for which the certificate is valid. However, there may be some situations when the certificate must abnormally be declared invalid prior to its expiration date. This is called certificate revocation. This can be viewed as "blacklisting" the certificate. This means that the existence of a certificate is a necessary but not sufficient evidence for its validity. A method for revoking certificates and distributing this revocation information to all the involved parties is thus a requirement in PKI. The reasons for revoking a certificate may be: suspected/detected key compromise, change of principal name, change of relationship between a principal and the CA (e.g. Alice may leave or be fired from the company) or end of CA's trust into the principle due to any possible reason.

The revocation mechanism should have an acceptable degree of timeliness, i.e., the interval between when the CA made a record of revocation and when this information became available to the relying parties should be small enough to be acceptable. Further, it is very important for the revocation mechanism to be efficient as the running expenses of a PKI derives mainly from administering revocation [4].

2 Available Revocation Techniques

This section briefly outlines a number of available revocation schemes-

2.1 Certificate Revocation Lists (CRLs)

CRLs are the most common and simplest method for certificate revocation. A CRL is a periodically issued list containing the certificate serial number of all the revoked certificates issued by a particular CA. This list is digitally signed by the CRL issuer to avoid tampering. The relying parties willing to validate a certificate issued by a particular CA can then download the most recent CRL of that CA.

Many variants of this "basic" CRL scheme have been designed to improve the performance. These include delta CRLs [2], partitioned CRLs and over-issued CRLs [3].

CRLs have been criticized for not being able to provide the required service and for being too costly [7, 8, 9, 11, 12]. We analyze the arguments advising against and for the use of CRLs in section 3.

2.2 Trusted Dictionaries

There are a number of schemes in which the end entities (relying parties) are supplied information in support of validating a single certificate rather than a complete list. A problem with this method is that the process of digitally signing each revocation reply is processing intensive. Trusted dictionary schemes attempt to solve this problem by using one-way hash functions in order to provide lightweight digital signatures. These techniques include Certificate Revocation Status Directory (CRS Directory) [7], Hierarchical Certificate Revocation Scheme [16], Certificate Revocation Trees [17], Naor's and Nissim's Scheme [10].

Common for most of these schemes are that they are not standardized like CRL and online schemes, and only CRT has been implemented for common use [18]. These schemes do not (with the exception of CRT) support the expressiveness found in CRLs. Additionally, these schemes are difficult to understand and implement as compared to CRLs and online mechanisms. These factors have limited the widespread use of trusted dictionaries.

2.3 Online Revocation Mechanisms

As a response to the low timeliness of some periodically updated certificate revocation schemes, protocols for online status checking have been developed. Many certificate based systems cannot tolerate the revocation delay resulting from the periodically updated schemes. With real time revocation checking, any party can confirm/obtain the proof of the certificate validity by performing an online transaction that indicates the current revocation status for a certificate.

We briefly summarize the common online revocation techniques-

2.3.1 On-Line Certificate Status Protocol (OCSP)

The OCSP [5] is a protocol developed by IETF in which on-line revocation information is available from an OCSP responder thorough a request/response mechanism. OCSP is designed to check the certificate revocation status exclusively.

The protocol is applied between a client (OCSP requester, acting for the user) and a server (OCSP responder, representing a directory). The client generates a so called OCSP request that primary contains one or even more identifiers of certificates queried for validity check, i.e. their serial number together with other data. Then, the (optionally signed) request is send to the server. The server receiving the OCSP request creates an OCSP response: The response mainly includes a timestamp representing the time when the actual request was generated, furthermore, the identifiers and status values of the requested certificates together with a validity interval. A certificate status value is either set to good, revoked or unknown. Be aware that "good" implies three meanings: firstly, the certificate is not revoked, but secondly, it may also not be issued yet or even thirdly, the time at which the response is produced is not within the validity of the certificate. Status "revoked" stands for a revocation or on hold of the certificate. If the answer is "unknown" the server has no information available about the required certificate. The validity interval specifies the time at which the status being indicated is known to be correct and optional the time at or before newer information will be available about the status of the certificate. The OCSP response should be digitally signed either by the server or by the CA. In case of any error the OCSP response contains an error message. The OCSP response is send to the requesting client of the user who then analyzes the data.

Depending on proper defined time schedules, OCSP provides more timely status information than any other method. A pre-producing of signed responses is currently optional. OCSP is especially appropriated for attribute certificates where status information always needs to be up-to-date.

2.3.2 OCSP-X, SCVP and DC

There are a number of on-line protocols that are more extensive than OCSP. OCSP-X [19], or OCSP extensions, provide a richer set of semantics for OCSP. With these extensions, an end entity is able to delegate the full task of deciding whether a certificate should be relied upon and whether it is acceptable for a particular operation.

The Simple Certificate Verification Protocol (SCVP) [20] is a separate protocol that is capable of handling (parts of or) the entire certificate validation process. With SCVP, end entities can avoid the overhead involved in processing the certificate validation locally. The protocol may also be used to centrally enforce some validation policy.

The Data Certification Server (DCS) [21] is a trusted third party that can be used as a component in building non-repudiation services. DCS is capable of verifying the correctness of specific data submitted to it. This service may, for example, be used to verify the correctness of a signature, the full certification path, and the revocation status of a certificate. Note that DCS provides more general services than OCSP-X and SCVP.

2.3.3 Obtaining new certificates

Ronald Rivest [11] criticizes CRLs and points out several design principles which cannot be fulfilled by CRLs. Rivest proposes an online approach in which if the most recent certificate fails to satisfy the recency requirements of the acceptor, the principal should obtain a more recently issued certificate from the CA. Hence, if Alice has a week old certificate indicating employment at the company and Bob is willing to accept at most a day old evidence of employment, Alice should query the online CA and get a new recent certificate created for her. Note that Alice may use this certificate again for other transactions.

The approach clearly has advantages i.e. the acceptor is able to set the recency requirements, certificate validation is reduced to just validating the digital signature on the certificate, acceptor need not deal with any revocation mechanism and better load distribution on the sender and the acceptor. A drawback is the increased load on the certificate servers. The certificate servers are now required to sign many more certificates than before.

3 Certificate Revocation Lists or Online Mechanisms?

While the approach of Trusted Dictionaries has not been deployed in common practice, choosing between CRLs and online mechanisms is still a sufficiently confusing task for PKI designers. Both CRLs and online mechanisms have their own advantages and disadvantages.

It has been argued [7, 8, 9, 11, and 12] at length that CRLs are both semantically and technically inferior to online mechanisms. There are a number of quite convincing arguments supporting this statement. We analyze the reasons from various paper as well as present some new reasons for the criticize of CRLs in the following propositions-

1. Recency Requirement must be set by the acceptor, not by the certificate issuer (CA). The reason is that the acceptor is the party who is running the risk if his decision is wrong, not the CA. Bob may want at most a day old evidence of employment at the bank before granting Alice the access to bank accounts of the customers. Weekly issued CRLs cannot meet his requirements. CRLs require the verifier to accept a recency guarantee bounded by the rate at which CRLs are generated.

2. The cost of CRL management and distribution is too high. Because of the potential size of CRLs, scaling to large communities can be difficult. To verify the certificate of Alice, Bob should download the complete CRL of the Alice's CA. The result of a recent simulation study [18] indicates that the maximum network load in case of CRLs is about 10 times higher than in case of online approaches.

3. CRLs are inappropriate for transactions that require real time revocation state information. That is, the inherent costs of CRLs generation and especially distribution prohibit online CRL generation.

4. For efficiency, the principal (sender) should supply all relevant validity evidence including recency information. More precisely, this states- *"For best load distribution, do work for your certificates yourself"*. There are several reasons for this proposition: - a) the sender can query the CA as well as the acceptor can, b) the recency information obtained may be useful again to the sender, c) this structure puts any burden on the sender (usually the client) rather than on a possibly overworked acceptor (the server). Even in cases, when the sender is the server (e.g. in https protocol, while establish an SSL connection, server sends its certificate), it is not much work for the server to query the CA and obtain a recent certificate daily (or even hourly). This approach is clearly better than having each client obtain the CRL of the server's CA to verify the server's certificate, d) in many case, this allows the acceptor (server) to be implemented in a stateless manner. For example, Bob can reply to Alice, "Sorry, please make sure that your evidences are at most one week old," and then forget about Alice until she comes back again, rather than having to rummage all over the Internet to see if Alice's certificates are still OK. A stateless server design is less vulnerable to denial-of-service attacks.

5. The distribution of request rates for the CRL distribution server is poor. If the weekly CRL is issued by the CA on Monday morning, clearly the request rate for CRLs will be much higher on Mondays and Tuesdays and will be quite low on Saturdays and Sundays. This high peak request rate shoots up the processing and network bandwidth requirements for the CRL server. The requests in case of online mechanisms are perfectly evenly distributed making them the most cost effective solution.
6. New certificates are the best evidence of recency. If a (new) certificate with a guaranteed validity period is available, then the acceptance process may be reduced to the validation of a single certificate signature. As the revocation state is implied by the existence of the certificate, CRLs are unnecessary.
7. Certificates in traditional CRL based schemes do not have any inherent recency information other than the certificate lifetime. Thus, each time a certificate is accessed, the verifier is required to obtain and validate a suitably recent CRL. Combined with proposition 7, this makes a strong argument for the use of online revocation mechanisms [5].
8. CRLs do not provide positive statements. Because CRLs only identify revoked certificates, the existence of a (non-revoked) certificate cannot be determined solely from the validity information.
9. Sometimes, downloading the CRL may introduce unacceptable latency in certificate validation. Since the acceptor should first download the most recent copy of the CRL of the sender's CA before validation (in case it doesn't have one), the delay introduced in the certificate validation may be significant.

These propositions give evidence of the problem with CRL based techniques and argue that CRLs should be eliminated in favor of online mechanisms.

While these arguments are definitely true and convincing, it should be noted that even after having of so many limitation and drawbacks, there exist some PKI environments where CRLs may still be the most cost effective revocation solution. [6] This is because of the two main reasons-

1. Though the bandwidth requirement for CRLs is clearly much higher than that for the online approaches, CRL based mechanisms avoid much of the cost associated with signature generation. Only one digital signature is periodically required by the CRL server. In contrast, online mechanisms place a heavy burden on the revocation server. This demonstrates a chief performance tradeoff between online and CRL mechanisms; CPU cost vs. bandwidth cost.

2. CRL are an attractive option in tightly coupled environments when reference locality is observed, i.e. when the acceptor has to validate many certificates issued by a single CA, periodically downloading the CRL of that CA may be a cheaper option.

A classical example when CRL are usually the best option is the Intranet Service. Certificate in a PKI running on an organization's Intranet are usually issued by a single/small number of CA's. Hence the relying (accepting) parties have the advantage of reference locality. Further, the bandwidth is not much problem as far as Intranet of any organization is concerned. Hence in this case CRLs are an attractive choice as far as real time revocation is not required.

So, we see that one must select the right PKI solution keeping in mind these points and the availability of resources. If real time revocation is required, online mechanisms are the obvious choice. For others, a careful choice should be made between CRLs and online approaches. The above points may serve as guidelines while selecting the revocation solution.

4 Conclusions

We briefly study the currently available revocation methods. Selecting the right revocation solution is important as the running expenses of a PKI derives mainly from administering revocation. While the approach of trusted dictionaries is limited, CRLs and online methods are commonly used as revocation methods. We study and compare these two approaches in light of their advantages and drawbacks in different environments. We conclude that online mechanisms are generally the most efficient vehicle for the distribution of the revocation information though CRLs should definitely be considered when the PKI environment offers reference locality and does not have bandwidth bottlenecks.

In past, most of the research has focused on creating new and more efficient revocation mechanisms. Now that there are a number of revocation options available, the problem of selecting the right revocation solution for the target environment assumes special importance. In this paper, we provide an analysis of various revocation options resulting in guidelines which one should keep in mind while selecting a revocation solution. These guidelines can greatly ease the task of a PKI designer as far as selecting the right revocation option is concerned.

References

1. A. Arsenault and S. Turner, PKIX Roadmap, Internet Draft, "Work in progress, IETF PKIX working group", October 1999.
2. Warwick Ford and Michel S. Baum, Secure Electronic Commerce, Prentice Hall PTR, 1997.
3. David A. Copper, A model of certificate revocation, proceedings of the Fifteenth Annual Computer Security Application Conference, December 1999.
4. Stuart Stubblebine, Recent-secure authentication: Enforcing revocation in distributed systems, In Proceedings 1995 IEEE Symposium on Research in Security and Privacy, pages 224-234, May 1995.

5. A. Malpani S. Galperin M. Myers, R. Ankney and C. Adams, RFC 2560: X.509 internet public key infrastructure online certificate status protocol - OCSP, June 1999.
6. Patrick McDaniel and Aviel D. Rubin, A response to "can we eliminate certificate revocation lists?", Financial Cryptography, pages 245-258, 2000.
7. S. Micali, Eficient certificate revocation, Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, March 1996.
8. J. Millen and R. Wright, Certificate revocation the responsible way, Post-proceedings of Computer Security, Dependability and Assurance: from Needs to Solutions (CSDA'98), IEEE Computer Society.
9. M. Myers, Revocation: Options and challenges, Lecture Notes in Computer Science, volume 1465, pages 165-171, 1998.
10. Moni Naor and Kobbi Nissim, Certificate revocation and certificate update, Proceedings 7th USENIX Security Symposium (San Antonio, Texas), Jan 1998.
11. Ronald L. Rivest, Can we eliminate certificate revocations lists? Financial Cryptography, pages 178-183, 1998.
12. Fox and LaMacchia, Certificate revocation: Mechanics and meaning, Financial Cryptography, LNCS, Springer-Verlag, 1998.
13. Carl A. Gunter and Trevor Jim, Generalized certificate revocation, Symposium on Principles of Programming Languages, pages 316-329, 2000.
14. R. Housley, W. Ford, W. Polk, and D. Solo, RFC 2459: Internet X.509 public key infrastructure certificate and CRL profile, January 1999. Status: PROPOSED STANDARD.
15. P. C. Kocher, On certificate revocation and validation, Financial Cryptography, LNCS, Springer-Verlag, 1998.
16. William Aiello, Sachin Lodha, and Rafail Ostrovsky, Fast Digital Identity Revocation, Advances in Cryptology - CRYPTO '98, Springer, 1998.
17. Paul Kocher, A Quick Introduction to Certificate Revocation Trees (CRTs), Technical report, ValiCert, 1999.
18. Andre Arnes, Public Key Certificate Revocation Schemes, Master's thesis, Department of Telematics, Norwegian University of Science and Technology, February 2000.
19. Phillip Hallam-Baker, OCSP Extensions, Internet Draft, "Work in progress, IETF PKIX working group", September 1999.
20. Ambarish Malpani and Paul Hoffman, Simple Certificate Validation Protocol, Internet Draft, "Work in progress, IETF PKIX working group", April 1999.
21. Carlisle Adams and Robert Zuccherato, Data Certification Server Protocols, Internet Draft, "Work in progress, IETF PKIX working group", September 1999.