FAST DIGITAL CERTIFICATE REVOCATION

An alternative to short lived certificates

Vipul Goyal

Department of Computer Science & Engineering, Institute of Technology, Banaras Hindu University, India. E-mail: vipulg@cpan.org.

Abstract: Digital Certificates are central to the concept of Public Key Infrastructures (PKI) and serve as a cryptographic proof of one's public key. Occasionally, certificates must be declared invalid prior to their due expiration date in case of key compromise or change in identity. Thus all PKIs should provide a mechanism through which an issued certificate may be revoked. The revocation mechanisms are commonly classified into Certificate Revocation Lists (CRLs), trusted dictionaries and online mechanisms. We briefly discuss the existing certificate revocation techniques and then present a new online revocation technique. More precisely, we present an alternative to short lived certificates proposed by Rivest.

Key words: PKIs, certificate revocation, online solutions, short lived certificates

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)

Y. Deswarte et al. (eds.), *Security and Protection in Information Processing Systems*

Certificates are tamperevident (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 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, over-issued CRLs [3], Blacklist CRLs and Redirected Pointers.

CRLs have been criticized for not being able to provide the required service and for being too costly [7, 8, 9, 11, 12]. See [22] for a comparative analysis of CRLs and online revocation techniques.

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. Instead of individually signing each revocation reply, trusted dictionary schemes attempt to solve the problem by using one-way hash functions in order to provide lightweight digital signatures.

A notable technique under this category is CRS. Micali introduced the certificate revocation status (CRS) scheme in 1995 and improved it in 1996 [7]. It was designed according to the following strategy: increase the amount of information transmitted by CAs to directories during an update, but design this update information in a way to enable directories to answer certificate revocation queries more succinctly.

CRS introduces two 100-bit values Y and N into a certificate, to represent "valid" and "revoked" status, respectively. These two values are calculated as followings: CA chooses a proper one-way hash function H, determines a lifetime period t and generates two 100-bit random secure values Y_0 and N_0 indicating valid and revoked, respectively. The above $Y = H^t(Y_0)$, $N = H(N_0)$. The scheme works as follows: To make revocation information up-to-date, during each time interval i (where $0 < i < t$), every CA submits the following information to its directories: an authenticated and time stamped 220-bit string L, signed by the CA, containing all serial numbers of issued and not-yet-expired certificates, and 100-bit value V for each certificate indicating whether it has been revoked or not within the current time interval (where $V = H^{t-i}(Y_0)$ for a valid certificate or $V = N_0$ for a revoked one). When a directory receives a certificate validity checking request, it sends proper value V to the verifier, based on which the verifier can check certificate validity.

Micali further revised his CRS scheme in 2002 [23]. The new scheme, called NOVOMODO, has one minor and one major modification: the minor one is that NOVOMODO uses SHA as the hash function, and the major one is that basic NOVOMODO discards directories at all. The author did describe how to build a distributed NOVOMODO, but update cost then is high, which still limits the scheme's scalability.

Aiello et al. [16] improved CRS approach by reducing update costs while maintaining its cheap query costs.

CRS and its variants again cannot provide timely revocation information although they are timelier than CRLs. The recency requirements are determined by the CA rather than the acceptor.

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)

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. 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

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 simply 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. This technique is also recommended for use in SPKI [25] and SDSI [24].

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. THE PROPOSED SOLUTION

In this section, we present a new online revocation mechanism based on time stamping the certificate serial number. Our technique is based on a variation of [11] and offers significant advantage over [11].

To explain our technique, we first introduce the concept of certificate renewal-

Certificate renewal is the certificate serial number together with timestamp (current date and time) digitally signed by the certificate renewal authority. A certificate renewal with serial number n and timestamp of time t is the proof that certificate having serial number n was not revoked and was still valid at time t.

Now, the mechanism proceeds as follows-

Step 1: The sender makes sure that his latest certificate renewal satisfies the recency criteria of the acceptor. If not, the sender obtains a new certificate renewal as follows

- The sender queries the certificate renewal authority (CRA) by just sending the serial number of his certificate.
- CRA checks the revocation status of the certificate having this serial number. If it is un-revoked, CRA creates the certificate renewal by digitally signing the time-stamped (with current time) serial number. Else, CRA timestamps the serial number with a time before the certificate creation time and signs it. This acts as a proof of *"Certificate Expiry"*.
- The CRA sends the certificate renewal (or certificate expiry) to the sender.

Step 2: The sender sends the certificate renewal and optionally, the certificate to the acceptor. Sending the certificate is not required if the acceptor already has a copy of the sender's certificate in its cache.

Step 3: The acceptor verifies the digital signature on the certificate (in case it was not cached) and the certificate renewal. It also makes sure that the certificate renewal satisfies its recency criteria.

First we present the advantage our scheme offers in general over other revocation mechanisms like CRLs and CRS.

1) Acceptor could simply reject a certificate renewal and ask the sender to obtain a more recent one. Thus, the acceptor is able to set the recency requirements. This is important since 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. Though CRS is better, even it does not satisfy this condition.

2) The bandwidth consumption is quite low contrary to CRLs. This is because individual proof of validity of a certificate is issued by the CRA in the form of certificate renewals rather than long CRLs. 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 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) Our method provides real time revocation status information. This is especially important in electronic commerce transactions. Note that CRLs and CRS cannot provide real time revocation status information.

4) The sender supplies all the relevant validity evidence, including recency, to the acceptor. More precisely, this means that the design principle-

"For best load distribution, do work for your certificates yourself"

is fulfilled.

There are several reasons for this principle

- The sender can query the CA as well as the acceptor can.
- The recency information obtained may be useful again to the sender.
- 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.
- 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.

Note that CRS also satisfies this design principle.

5) The distribution of requests for the CRA is uniform in our technique. However the distribution of requests 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 than on Saturdays and Sundays. This high peak request rate shoots up the processing and network bandwidth requirements for the CRL server. Even in the case of CRS, the distribution of requests is not uniform.

6) 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. Our solution does not require the verifier to contact any third party during the certificate validation process. Thus the delay introduced is limited to the delay due to signature verification.

All the above advantage our technique offers are also possible by using short lived certificates as proposed in [11]. We now proceed to compare our scheme specifically with [11].

1) Network Load

The network load in our revocation scheme is significantly lower than in [11]. The bandwidth consumption is reduced for at least two of the three parties involved (i.e. the sender and the CRA). Additionally, the bandwidth consumption by the acceptor may also be reduced if it can cache the certificates sent to it.

Instead of sending a complete new certificate, the CRA now sends just a certificate renewal to the sender which is significantly smaller in size as compared to a new certificate. Further, in our scheme, the certificate does never change in contrast to [11]. This allows the caching of certificate by the acceptor. Hence the sender does not have to send the certificate now to the acceptor if it is cached.

The average network load with our technique (assuming 1 KB X.509 certificates) will be around 65 % assuming no caching, 40 % assuming 50% cache hit rate and 15 % assuming 99% hit rate as compared to [11].

2) Latency

Since the data to be transferred over the network is reduced for each of the three parties, the latency involved in the communication is also reduced.

3) Computational Requirements for the Parties involved

a) For the sender, the processing load is the same (almost negligible) for our technique as well as [11].

b) For the CRA, the overhead of creating a new certificate is eliminated as well as the data to be signed is reduced.

c) The processing load analysis for the acceptor yields the following-

- When the certificate is not cached:- The acceptor has to verify two digital signatures (certificate signatures and certificate renewal signatures) as compared to one in [11]. Hence, this is a drawback of our approach. However if the signatures schemes like RSA are employed, where signature verification is much faster than signature generation, this drawback may not be very significant.

- When the certificate is cached:- The acceptor has to verify only one digital signature (certificate renewal signature). However, an advantage over [11] is that the data to be verified here is much smaller and the certificate content check and validation is not required. Certificate content check includes verification and validation of various certificate fields like validity period, identity of the holder, public key etc.

4) Storage requirements for the CRA Server

In [11], since the online certificate generation is required, every certificate (or its fields, such as public key and other attributes) should be stored on the server. In our approach, the CRA doesn't need to deal with or even store certificates on its server. Just a database of serial numbers of revoked certificates is sufficient. While signing a certificate renewal for the queried serial number, the CRA just checks for its presence in the database of revoked serial number. If the serial number is not present in the database, it can be inferred that the certificate in question is un-revoked and thus the CRA proceeds to sign the certificate renewal.

This kind of design also offers another interesting advantage. Consider an organization in which certificates are exchanged within the organization and the information specified in the certificates is considered sensitive due to privacy concerns/ some other reasons. Hence this information should be prevented from leakage outside the organization. Storing all the information specified in the certificates on an online server presents an adversary a prime target of attack through which she may obtain all the desired information. However, our technique defends this kind of attack since no certificate information is stored at the server at all.

5) Security of private key of the Certificate Authority

CAs may wish to avoid placing their private key on hosts connected to Internet. However, since [11] required online certificate creation, CAs are forced to do so. This could create mission critical components in the server

security design as any compromise of CAs private key may lead to the catastrophic failure of the PKI. Our approach does not require online certificate creation. The keys for certificate creation and certificate renewal could be different. We only require the certificate renewal key to be placed online, which if compromised, would enable the attacker to renew revoked certificates but would not enable her to create new certificates.

6) Clean Separation

With a clean separation between certificate creation and certificate revocation (using certificate renewals), it becomes possible for the CAs to delegate the revocation process. Such a possibility is of great interest in today's world. Organizations nowadays frequently outsource a part of their work to other entities. Hence CAs may choose to stick to the process of certificate creation and outsource the whole revocation process.

Further, with such a clean separation, the failure (key compromise) of revocation system does not imply the failure of certificate creation system and vice versa.

Key compromise Issue-

Rivest [11] proposes that key compromise is different and the issue should be handled differently. It proposes Key Compromise Agents who will form a high speed reliable network among themselves and would issue a certificate of health to each principle. Certificate of health is the proof that the key hasn't been compromised yet. [11] suggests that a key compromise should revoke the certificate of health rather than the ordinary certificate.

To analyze this proposal, we come back to the basic guarantee of a certificate-

"We, the CA, declare that the public key of Alice is ..."

Implied guarantee-

"Any message encrypted with this key will be readable by Alice only. Further, only Alice can produce digital signatures verifiable with this key".

A key compromise obviously renders this certificate guarantee invalid. Further, after the key compromise, the certificate does not seem to be of any use to Alice as she will be probably changing her keys and won't be using public key specified in the certificate anymore. The most obvious solution in this case appears to be the revocation of the certificate. This will eliminate the need of expensive Key Compromise Agents and certificates of health. Hence, we choose to stick with Alice just informing her CA or the certificate renewal authority about the key compromise which would then revoke her certificate and issue a fresh certificate with her new public key on it. This seems to be logically consistent as well as the most efficient solution.

4. CONCLUSION

We proposed a new online revocation scheme based on certificate renewals. Our scheme presents an alternative to short lived certificates. Our scheme offers advantages in terms of network load, communication latency, computational requirements, CRA server storage requirement and CA key security. A clean separation between certificate creation and certificate revocation permits CA's delegation/outsourcing of the revocation process.

We propose that the current notion of digital certificates (e.g. X.509) be extended to include the certificate renewal in an extension field of the certificate itself. This field will keep changing and will be excluded while signing the certificate. This will perhaps give a cleaner and easier to think Public Key Infrastructures with all revocation information embedded in the certificate itself.

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. In 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?". In 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. In Post-proceedings of Computer Security, Dependability and Assurance: from Needs to Solutions (CSDA'98). IEEE Computer Society.

[9] M. Myers. Revocation: Options and challenges. In Lecture Notes in Computer Science, volume 1465, pages 165-171, 1998.

[10] Moni Naor and Kobbi Nissim. Certificate revocation and certificate update. In Proceedings 7th USENIX Security Symposium (San Antonio, Texas), Jan 1998.

[11] Ronald L. Rivest. Can we eliminate certificate revocations lists? In Financial Cryptography, pages 178-183, 1998.

[12] Fox and LaMacchia. Certificate revocation: Mechanics and meaning. In Financial Cryptography. LNCS, Springer-Verlag, 1998.

[13] Carl A. Gunter and Trevor Jim. Generalized certificate revocation. In 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. In International Conference on Financial Cryptography. LNCS, Springer-Verlag, 1998.

[16] William Aiello, Sachin Lodha, and Rafail Ostrovsky. Fast Digital Identity Revocation. In 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.

[22] Vipul Goyal, "Certificate Revocation Lists or Online Mechanisms", Workshop on Security in Information Systems, Portugal, April 2004.

[23] Silvio Micali. NOVOMODO: Scalable Certificate Validation and Simplified PKI Management. In 1st Annual PKI Research Workshop - Proceeding, April 2002.

[24] Ronald L. Rivest and Butler Lampson. SDSI - a simple distributed security infrastructure. (see SDSI web page at http://theory.lcs.mit.edu/~cis/sdsi.html).

[25] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "SPKI Certificate Theory", RFC 2693, September 1999.