Digital Certificates: A Survey of Revocation Methods

Petra Wohlmacher
University of Klagenfurt
Villacherstrasse 161
9020 Klagenfurt, Austria
+43 463 2700854
petra.wohlmacher@uni-klu.ac.at

ABSTRACT

Digital certificates form a basis that allows entities to trust each other. Due to different constraints, a certificate is only valid within a specific period of time. Coming from several threats, there are important reasons why its validity must be terminated sooner than assigned and thus, the certificate needs to be revoked. This paper provides a classification of revocation methods and gives an overview of the main methods like CRL, CRS, CRT, and OCSP. If and in which way a revocation method is suited must be analyzed in accordance to their purpose.

Keywords

Digital certificate, public-key certificate, attribute certificate, X.509, revocation, CRL, CRS, CRT, OCSP.

1. INTRODUCTION

Nowadays, large security infrastructures are developed and currently going to be established for meeting security requirements [11]. Different areas are using these infrastructures to enhance the security of their IT-systems, their applications, and also the communication between different entities (like users, institutions, processes or devices). Security is defined by security requirements, e.g. confidentiality, integrity, authenticity, and non-repudiation, which are met by security measures. Most of these measures use cryptographic mechanisms, e.g. ciphers, digital signatures, and authentication protocols, but also access control mechanisms. The security of these measures substantially depends on the authenticity of specific data like public keys and sometimes even attributes. Both data need to be linked to its owner in an authentic manner. Such a link can be provided by public-key certificates and attribute certificates, respectively [4]. Certificates are particularly best for a use in interconnected open systems. They are suited for applications, where a large number of entities exists and even entities associated to different institutions unknown to each other have to authenticate and communicate securely. Here, certificates form a basis that allows entities to trust each other.

Certificates are generated and issued by a trustworthy authority named certification authority (CA). The security policy of a CA describes the lifecycle of key pairs or attributes, respectively, and thus, the validity period of a certificate. Within this period, its reliability can be assured. The validity of the public key or attribute is specified in the certificate and signed together with other data by the CA. Therefore, certificates are unforgeable and usually securely submitted to an authority that provides certificates. The authority is mostly called directory. Typically, the validity period of a certificate is between several months and two years. But in some circumstances, a certificate must to be revoked, i.e. its validity must be terminated sooner than assigned.

The revocation management needs to be clearly defined for CAs, directories, and users: CAs must provide a revocation service in a trustworthy manner and therefore, publish a proper security policy. A user needs to know how and when a revocation must be initiated and also gets informed. The revocation is initiated by the owner of the certificate (subject), by an authorized representative which is already mentioned in the certificate or by a CA. Only the CA revokes certificates and complies with a revocation request since the initiator is able to prove his authorization. Usually, the status of all certificates is submitted to the directory that answers users requests concerning the validity of certificates. Due to the security policy, this service might also be provided by another authority.

Additionally, revocation methods must fulfill other requirements, too. A revocation needs to be fast, efficient, timely and particularly appropriated for large infrastructures. Due to that, it is necessary e.g. to reduce the number of time-consuming calculations like verification processes of a digital signature and to apply other mechanisms, or to minimize the amount of data transmitted. It is also desirable that a method provides suspending a certificate temporarily (placed onhold) and also a reuse.

To prove the validity of a certificate, a user has to perform different tests, where some of them are really critical. One of the most critical ones is to determine whether a certificate has been revoked or not. Usually, a user determining whether a specific certificate has been revoked sends a request to a directory. The request contains at least a serial number which represents a unique identifier for each certificate. The response includes the serial number, status, date and reason of revocation, and is then analyzed by the user.

In the following, the paper gives an overview of different kinds of revocation methods. They all have in common that an authentic verification key of the (Root-)CA is required. Initially, we give a short classification and point out main reasons for the revocation of certificates. Traditionally, revoked certificates are stored in certificate revocation lists (CRL) which are described in section 4. Section 5 gives an introduction to certificate revocation systems (CRS). In section 6 we present the idea of a certificate revocation tree (CRT) and in section 7 we point out the online certificate revocation protocol (OCSP). Finally we give some conclusions.

2. CLASSIFICATION

Methods for revocation can be classified in different ways:

1. By the way of checking: The check can be performed either offline or online, sometimes both methods are applied. Within an offline scheme, the validity information is precomputed by a CA and then distributed to the requester by an non-trusted directory. Within an online scheme, the status information is provided online by a trusted directory. A proof of validity is performed during each request and provides up-to-date information.
2. By their kinds of lists: Negative (black) lists contain revoked certificates and positive (white) lists contribute valid certificates. Sometimes both mechanisms are combined.
3. By the way of providing evidence: A direct evidence is given if a certificate is mentioned in a positive or negative list, respectively. Then it is supposed to be not revoked or revoked, respectively. An indirect evidence is given, if a certificate can not be found on a list and therefore, the contrary is assumed.
4. By the way of distributing information either via a push or pull mechanism.

3. REASONS FOR REVOCATION

Coming from several threats, there are important reasons why a certificate needs to be revoked [4]:

- Key compromise: The private key of the subject (user) or of the issuer (CA) has been compromised or is suspected to be compromised (e.g. broken or stolen).
- Change of affiliation: Some information in the certificate about the subject or any other information is not longer valid.
- Superseded: The certificate is superseded - no further reasons are made available.
- Cessation of operation: The certificate is not longer needed for its assigned purpose.

Additionally, there are some further arguments why a certificate needs to be revoked (descendently sorted by their urgency):

- Algorithm compromise: The signature algorithm used by the CA has been broken in general or the algorithm of the certified public key is compromised. This might be caused by new advances in algorithm theory, number theory or computer capabilities.
- Revocation of superordinated certificate: A certificate being part of the certification path is revoked.
- Loss or defect of security token, loss of password or PIN: Either the subject of the certificate has lost its physical equipment or its equipment is damaged. Regularly, a password or a PIN protects the token from unauthorized access and can be lost, too.
- Change of key usage: The certified key can not longer be used for its assigned purpose.
- Change of security policy: The CA does not longer work under its defined policy, e.g. it ceases to support a service for certificates.

Usually, the status information about the certificate include reasons for the revocation.

4. CERTIFICATE REVOCATION LIST

Certificate revocation lists (CRL) together with X.509 certificates have been introduced in 1988 by ITU-T (formerly CCITT). Since the second edition of the X.509-Recommendation in 1993, revocation lists are based on an improved version 2 by ITU-T and ISO/IEC [4]. A CRL represents a negative list giving indirect evidence and is provided offline. CRLs are periodically issued, usually monthly.

A CRL contains a list of serial numbers of revoked certificates together with their date of revocation, and also a date of its generation and a latest date of the next issue. Optional more information e.g. reasons of revocation can be added. Finally, the CRL is digitally signed by the issuing CA. Thus, its freshness and authenticity can be checked. CRLs are periodically sent to a directory.

Users requesting the validity of a certificate receive a full CRL. Then they check the actuality and verify the signature of the CRL. If this succeeds, they determine whether the certificate queried is included in the CRL or not. If the serial number can not be found, the certificate is supposed to be still valid.

Because CRLs are straightforward, they are easy to understand and thus, widespreadly used. Since the validity period of certificates is long and the number of users is immens, CRLs can grow extremely large. Therefore, a great amount of data needs to be transmitted. The fact that a CRL is only up-to-date at their point of issuing led to the definition of so called delta-CRLs. A delta-CRL is issued between two CRL updates. It includes only changes since the last issued CRL and so enhances the efficiency. Delta-CRLs contain sequence numbers that allow to verify the completeness of CRL information.

5. CERTIFICATE REVOCATION SYSTEM

The certificate revocation system (CRS) [8] has been introduced by Silvio Micali in 1995. His idea uses online/offline signatures [1]. He improved his idea in 1996 [9], where he redefines CRS by revocation status, and also gets a patent in 1998 [12]. A CRS mixes positive and negative lists and thus, gives direct evidence. The validity status of each certificate is treated separately. Here, a user sending a query concerning the validity of a single certificate, will get a response containing an individual, short information about this certificate. Depending on the up-to-date-time schedule, the system can either operate online or offline. In the following, we point out the main concept of a CRS [8].

The system is set up as follows: The CA defines n time intervals i (e.g., with respect to a year, daily: $n = 365$ and i represents a day), within the CRS is periodically updated. Using X.509 certificates, the number of extension fields needs to be extended by two 100-bit fields called Y (for "yes") and N (for "no"). Because of CA's signature, the authenticity of both values is guarantied. The CA constitutes a proper hash function H and chooses (pseudo-)randomly two 100-bit values Y_0 and N_0, where both Y_0 and N_0 are kept secret by the CA. Then the CA calculates (see Figure 1): $Y := Y_n = H^n(Y_0)$ and $N := H(N_0)$. Value Y_0 is used within n computations but N_0 only once.

To keep the CRS up-to-date the CA submits the following information to a directory: a fresh and timestamped list L containing all serial numbers of issued and not-yet-expired certificates, where L is signed by the CA. Also further information are transmitted: new certificates issued within interval i; a 100-bit value V for each certificate determined either by $V := Y_{n-i} = H^{n-i}(Y_0)$ if the certificate is neither expired nor revoked, or by $V := N_0$ if the certificate has been revoked within interval i. For revoked certificates the CA may also provide a signed template including additional data like time and reason of revocation. Now, the directory stores the serial numbers of each certificate together with its dedicated value V.

A user asking for the validity of a certificate first gains list L. Then he checks the soundness and correctness of the whole list L by verifying the signature. If this succeeds, he determines whether L contains the requested serial number. Further tests are performed by using Y or N (see Figure 1): He calculates $H^i(V)$ and examines whether $H^i(V)$ equals Y. If this is true, the certificate is valid within the interval i. Otherwise, he computes $H(V)$ and verifies whether $H(V)$ is equal to N. If any verification succeeds, the status of the certificate can determined. This works because: $Y = H^i(H^{n-i}(Y_0)) = H^n(Y_0)$ and $N = H(N_0)$.

All other occurrences come from problems concerning e.g. data transmission, data authenticity or even denial of services.

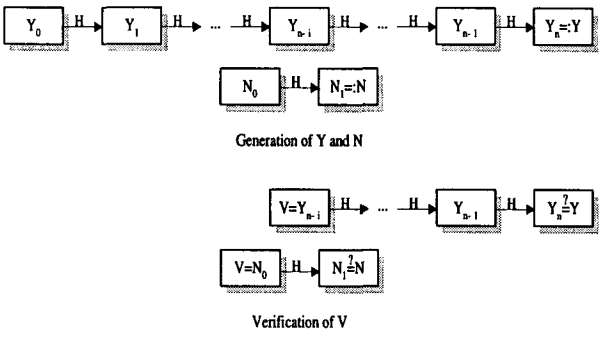

Figure 1: Certificate Revocation System

A CRS provides the following advantages: The signed list L is offered off-line. Because a hash function is used and both Y and N are represented by a string of 100 bits, the verification process of V is efficient and therefore, can be calculated online. The directory is not able to forge neither L nor V since Y_0 and N_0 is only known by the CA. Nevertheless, the security of CRS depends also on the secrecy of Y_0 and N_0, and also on their generation process.

6. CERTIFICATE REVOCATION TREE

Certificate Revocation Trees (CRT) have been introduced by Paul Kocher in 1998 [5] [6] and are based on hash trees [7]. CRTs are negative lists but also support information about non-revoked certificates (mixed form): Regarding the sorted set of revoked certificates, all still valid certificates can be assigned to specific validity intervals. Thus, they give direct evidence.

The system is initialized as follows: Let *low* resp. *high*, where $low < i < high$, determine the lower resp. upper bound of the range of all serial numbers i. A certificate with serial number i is named C_i. Revoked certificates C_j and C_k form a pair (j, k), where no certificate C_m with a serial number in the range $j < m < k$ is revoked. Let N be the number of revoked certificates, then the ranges are identified by data structures $L_0, \ldots, L_N$, where each of them may contain additional information about reason and date of revocation. Now, each $L_n (0 \leq n \leq N)$ is used as a leaf node $N_{0,n}$ of a binary tree to build a hash tree by use of a hash function H: $N_{0,n} = H(L_n)$. (Remark: To simplify matters, we assume that $N+1$ is a power of 2, where the binary tree is complete and has height $log_2(N+1)$. Otherwise, L_i is placed to lower levels of the tree.)

Each node $N_{i,j}$ of the next level (descendant) is computed by hashing the concatenation of its left ancestor $N_{i-1,l}$ and its right ancestor $N_{i-1,r}$: $N_{i,j} = H(N_{i-1,l}||N_{i-1,r})$, where H denotes a hash function. All other values of the nodes are computed in the same manner up to the root $N_{r,0}$ where $r := log_2(N+1)$. Subsequently, the root of the tree together with other information like issuing and expiration date of the CRT is digitally signed by the CA. Finally, tree and signature are made available for users by a directory.

A user sends a request containing the serial number of the certificate to a directory. The response consist of the following data: the data structure L_k which includes the inquired serial number, if k is even: the value $N_{0,k+1}$ otherwise $N_{0,k-1}$, additionally, the smallest number of other hash values representing nodes which are needed to compute the root and finally, the root and its signature. Now, the user needs to hash the data in the right manner and checks whether the computed value of the root equals its submitted value. If so, the validation succeeds and regarding L_i either the certificate is valid or revoked.

Figure 2 shows an example for a CRT, where $N = 7$ and the serial numbers of revoked certificates are given by 4, 8, 15, 16, 28, 34, 48. For example, $N_{1,2}$ is computed by $N_{1,2} = H(N_{0,4}||N_{0,5})$ where $N_{0,4} = H(L_4)$ and $N_{0,5} = H(L_5)$. The validity of certificate with serial number 14 can be checked using L_2, $N_{0,3}$, $N_{1,0}$, $N_{2,1}$, and the signed root.

CRTs are efficient, because of the use of hash functions and the amount of data increases only as the logarithm of the number of tree leafs. Furthermore, values of the nodes can be precomputed. The signing of the root can also be performed off-line, but since it is an off-line system, issuing dates need to be defined and also up-to-date problems occure.

7. ONLINE CERTIFICATE STATUS PROTOCOL

Another method is the Online Certificate Status Protocol (OCSP) [10] developed by IETF. It specifies a protocol used to determine the current validity status of a certificate on-

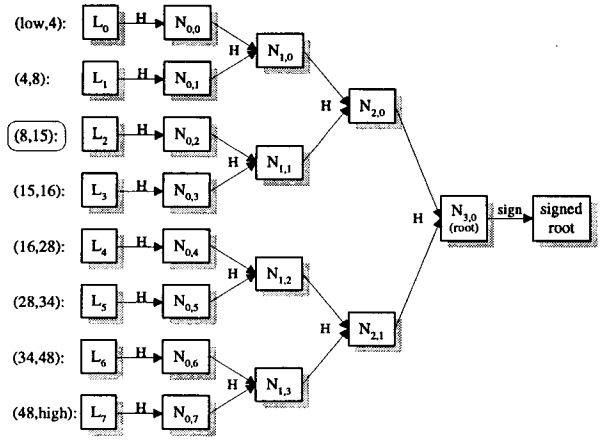

Figure 2: Example of a CRT

line. OCSP is designed for X.509 certificates but may also work with other kind of certificates. The protocol can be used instead of or even together with CRLs if more timely information about the status is required. Information about the way to obtain a certificates status can be included within the extension fields of a X.509-certificate.

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 ore even more identifiers of certificates queried, 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: Since all syntactical and content checks succeed, the response mainly includes a timestamp representing the time when the actual request is 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 onhold 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.

Extensions like time and reason for revocation may be used in addition, further OCSP extensions are handled in a separate Internet-Draft [2]. Formats of request and response are due to the transmission protocol e.g. HTTP or LDAP.

Depending on proper defined time schedules, OCSP provides more timely status information than any other method. A preproducing of signed responses is currently optional. OCSP is especially appropiated for attribute certificates where status information always need to be up-to-date. In the practice, the caching of HTTP-browsers must be handled carefully.

8. CONCLUSION

Regarding revocation of certificates different methods have been developed. Beside the presented methods, further methods exist. If and in which way a revocation method is suited must be analyzed in accordance to their purpose. An important aspect for a decision is its costs. High costs derive from a great amount of transmitted data that is needed to provide a proper revocation, but also from measures to provide the availability of timely data. Using offline systems, commonly the time period between two updates is long and therefore, the validity cannot be assured exactly. However, this is sufficient for the purpose of some applications. Online systems appropriated for purposes where more timely information is needed are obviously more expensive than an offline system. Another aspect is also whether a revocation method is applicatively for a storage equipment like smart cards or other security tokens.

The knowledge about different revocation methods is not very widely spread. Efficient and practicable methods are still needed and a topic of today's research. A main requirement for new developments and new ideas is that they can easily be integrated in widespreadly used X.509 certificates.

9. REFERENCES

[1] S. Even, O. Goldreich, S. Micali: On-Line/Off-Line Digital Signing. Proc. of CRYPTO 89, pp. 263–275.

[2] P. Hallam-Baker: OCSP Extensions. Internet Draft draft-ietf-pkix-ocspx-00.txt, Sep 03, 1999.

[3] P. Hallam-Baker, W. Ford: Internet X.509 Public-Key Infrastructure – Open CRL Distribution Process. Technical Report, IETF, 1998.

[4] International Telecommunication Union: ITU-T Recommendation X.509 (1997 E): Information technology – Open Systems Interconnection – The Directory: Authentication Framework, 6-1997 (also published as ISO/IEC International Standard 9594-8).

[5] P. Kocher: A Quick Introduction to Certificate Revocation Trees. http://www.valicert.com/resources/bodyIntroRevocation.html

[6] P. Kocher: On Certificate Revocation and Validation. Proc. of Financial Cryptography 1998, pp. 172–177.

[7] R. Merkle: Secrecy, Authentication, and Public-Key Systems. PH.D. Dissertation, Department of Electrical Engineering, Stanford University, 1979.

[8] S. Micali: Enhanced Certificate Revocation. Technical Memo MIT/LCS/TM-542, 1995.

[9] S. Micali: Efficient Certificate Revocation. Technical Memo MIT/LCS/TM-542b, 1996.

[10] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. IETF, Jun. 1999. http://www.rfc-editor.org/rfc/rfc2560.txt

[11] Public-Key Infrastructure (X.509) (pkix) http://www.ietf.org/html.charters/pkix-charter.

[12] United States Patent Nr 5.793.868: Certificate Revocation System, Silvio Micali, Date of Patent: 11. Aug 1998.