On Certificate Revocation and Validation

Paul C. Kocher

Chief Scientist, ValiCert
3160 Bayshore Rd., Palo Alto, CA 94303, USA.
e-mail: pck@cryptography.com.

Abstract. Cryptosystems need to check whether the certificates and digital signatures they are given are valid before accepting them. In addition to providing cryptographically secure validity information, certificate revocation systems must satisfy a variety of challenging technical requirements. The traditional revocation techniques of Certificate Revocation Lists (CRLs) and on-line checking are described, as well as a newer technique, Certificate Revocation Trees (CRTs), based on Merkle hash trees. CRTs provide an efficient and highly-scalable way to distribute revocation information. CRT-based systems include Tree Issuers who compile revocation information, Confirmation Issuers who distribute elements from CRTs, and users who accept certificates. CRTs are gaining increased use worldwide for several reasons. They can be used with existing protocols and certificates, and enable the secure, reliable, scalable, and inexpensive validation of certificates (as well as digital signatures and other data).

1 Brief Introduction

Industry is embracing public key cryptography and certificates as a practical and secure way to express privileges, relationships, and trust in a digital form. Without revocation support, however, many of the advantages of public key cryptography are lost.

For example, consider code signing. Before running a digitally signed program, the operating system might verify that the code has a signature from someone with a developer certificate. Without revocation checking, however, any signature made with any certified application developer's private key will be accepted. As a result, the compromise of any one developer's private key compromises the entire system unless users are able to reject signatures made with the compromised key. Without revocation, the security model is little better than having only one signing key and giving copies to all application developers.

Certification Authorities (CAs) and others who issue certificates and digital signatures are granting privileges, usually to a person, computer, or some data. In any large system, some digital signatures will be misused or need to be revoked. Secure revocation checking is essential because, unlike passports and other forms of physical identification, a compromised private key can be copied easily.

2 Technical Requirements

Implementing a good revocation system is difficult. The revocation status of any certificate (or signature, etc.) can change at any time, so certificate acceptors have to obtain fresh information quickly enough that attacks can be stopped. While specific requirements vary somewhat, a typical revocation system must provide the following:

Security Users must receive cryptographic assurance as to whether or not certificates are valid. An attacker should not be able to make a user accept a revoked certificate or reject a valid certificate as revoked. If required, the time and reason of revocation should also be provided securely.

Reliability The service must be available at all times, even if individual network components fail or are attacked. Reliability is essential for security so that a denial of service attack cannot be used to cause applications or users to accept bad certificates.

Scalability In a large system, such as secure e-mail on the Internet, revocation information must be provided from thousands of CAs to millions of users conducting billions of transactions, all at low cost.

Performance The entire system must be fast and efficient, without requiring excessive computational complexity or network overhead for any participant.

Memory Validation often has to be performed in constrained environments. Smartcards, for example, often can have less than a kilobyte of available RAM.

Bandwidth Communication bandwidth should be small and should scale well. By minimizing the amount of data that has to be exchanged, users experience shorter validation delays, less burden is placed on the network infrastructure, and the cost of providing the service is reduced.

Auditability While the requirements for the actual revocation decision itself vary widely between implementations, the cryptographic operations should be auditable.

Practicality The system must be easy to integrate into existing protocols and applications.

Manageability It must be possible to operate the system in a secure manner. Whenever possible, critical keys should be stored off-line where they are least vulnerable to attack.

Simplicity The cryptographic design must be unambiguous and easy to review and evaluate.

3 Certificate Revocation Lists

The traditional revocation technique is for CAs to issue Certificate Revocation Lists[1] (CRLs), which are digitally signed lists of the serial numbers of the certificates revoked by a CA. To test the validity of the certificates in a chain, a user

obtains current CRLs from each CA in the chain and checks for the certificates' serial numbers. To be secure, CRLs must be issued regularly and frequently so that users can reject stale CRLs. Otherwise attackers could exploit compromised keys by preventing victims from receiving fresh CRLs. (The CRL expiration period determines the time required to halt the use of revoked certificates.) Users must thus download a new CRL with each transaction, unless a fresh CRL is cached.

Certificate revocation is not widely used today because CRLs fail to meet many of the needs listed above:

Reliability The CRL distribution mechanism is not standardized, and depends on the CA and the application. While big commercial CAs can issue CRLs on a predictable schedule and operate reliable servers, smaller CAs (such as companies issuing certificates to their employees or participants in systems with non-hierarchical trust models) often cannot.

Scalability A CRL's size is proportional to the number of revoked certificates, which in turn is some fraction of the total number of issued certificates. A large system can easily have many thousands of revoked certificates. CRLs of many megabytes are possible, particularly if revocation occurs often. Techniques such as CRL segmentation can help somewhat, but do not solve the problem.

Performance Validation performance is likely to be slow, since users have to make separate network connections to obtain the CRLs from each CA in a chain, download the CRLs, and verify each.

Memory Because CRLs do not have a maximum size, they often cannot be processed by smartcards or in other constained environments.

Bandwidth CRLs can require very large amounts of bandwidth. For example, the CRLs needed to validate an S/MIME message will often far exceed the size of the message. Bandwidth requirements for CRL distribution alone can potentially exceed a network's entire capacity.

Practicality Today, certificates seldom specify if CRLs are available or where they can be downloaded. As a result, it is often impossible for applications to find CRLs.

4 On-line Verification

Revocation systems can be on-line or off-line. In off-line systems (such as CRLs and CRTs), validity information is precomputed then distributed. In on-line schemes, validity proofs are constructed for each request, such as by a digitally signing a certificate identifier, the certificate status, and a user-specified challenge value (to prevent replay attacks).

On-line schemes are much more expensive to operate than CRT- based systems because the signer is involved in every secure transaction where validation

is required – potentially requiring computing resources and tremendous bandwidth. Also, key management is difficult because the validation keys must be stored securely yet be connected to an untrusted network. However, true on-line schemes have the advantages that status information is always up-to-date and they are simpler to design and implement.

5 Background: ValiCert

My involvement with the revocation problem started with trying to design a secure electronic financial transaction system. The encryption algorithms, certificate issuance processes, etc. were available, but the system could not be built because existing certificate validation technologies were unworkable. My work on the problem led to the development of the (currently patent-pending) Certificate Revocation Tree (CRT) technology [2].

ValiCert, Inc. was founded to implement solutions to the revocation problem using CRTs and other technologies. Today ValiCert has developed a great deal of experience addressing real-world revocation requirements and is the only company in the world specializing in certificate validition and revocation.

6 Introduction to Certificate Revocation Trees

Certificate Revocation Trees (CRTs) are designed to solve the problems associated with CRLs and other revocation approaches.

The basic technology is best outlined with an example. Imagine a simple case where CA "Q" has revoked its certificates with serial numbers 5, 12, 13, 15, 20, 50, and 99. To build a CRT, the information from the CRLs is divided into a series of ranges that describe certificate statuses. For example, the range (5,12) for CA Q denotes "certificate 5 is bad, but any certificates larger than 5 and less than 12 are good." There exactly one range that describes each possible certificate serial number. For example, the range for certificate 12 would be (12,13), which means "certificate 12 is bad, but any certificates larger than 12 and less than 13 are good."

The ranges are packaged (with the reason for revocation, date of revocation, etc.) as data structures $L_0 ... L_{N-1}$, which are then used as leaf nodes to build a Merkle hash tree[3]. The tree's root and accompanying information such as when the tree expires are digitally signed (for example by ValiCert). Figure 1 illustrates the structure of a hash tree for the example above. In the hash tree, each node $N_{i,j}$ is formed by concatenating the values connecting from its left. For example, $N_{2,1}$ equals the hash of $N_{1,2}$ and $N_{1,3}$.

The tree and digitally signed root are distributed to Confirmation Issuers, servers that provide validation data to users. A user wishing to validate a certificate chain sends a data structure identifying the issuer and serial number of each certificate to any Confirmation Issuer. The Confirmation Issuer responds with the Merkle tree leaf for each certificate, the hashes that bind the leaves to the root, and the signed root. No cryptographic operations are required in this

process, as the tree's intermediate nodes can be precomputed. Thus, a single Confirmation Issuer can easily process millions of transactions per day.

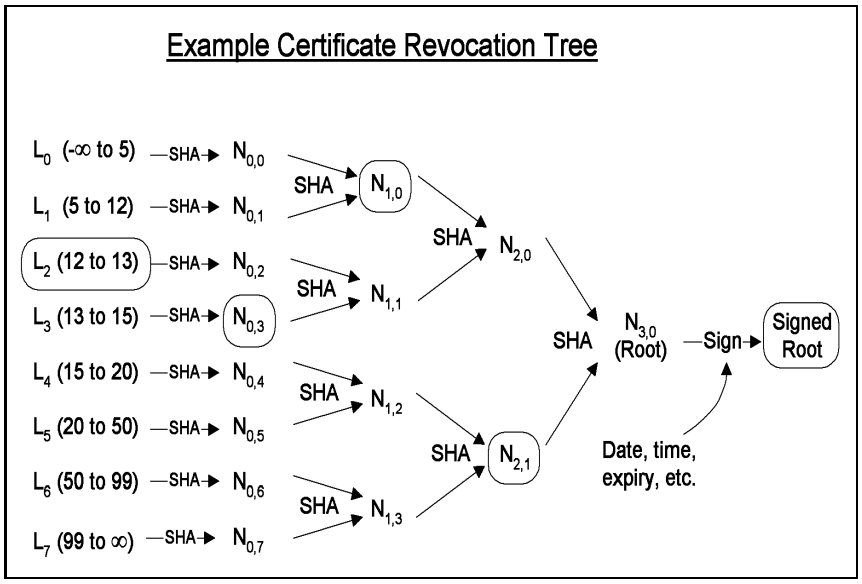

Figure 1: Sample Certificate Revocation Tree

The user confirms that the leaves describe the certificates in question, that the hashes correctly bind the CRT leaves to the root, and that the root is properly signed and not expired. Additional information, such as the date and reason for revocation, are also provided in the leaf nodes. As necessary, the user may also validate the root key (e.g., by verifying that it has been signed by the CA as authorized to provide validity status).

In Figure 1, the elements of the confirmation for certificate 12 are circled. The range L_2 is valid because it includes the correct serial number (i.e., indicates that 12 is revoked). Using the hash algorithm (e.g., SHA[4]), the user can derive $N_{0,2}$ from L_2. With the circled value $N_{0,3}$, the user can find $N_{1,1}$ by hashing the concatenation of $N_{0,2}$ and $N_{0,3}$. Next, $N_{2,0}$ can be found from $N_{1,0}$ and $N_{1,1}$. Finally, $N_{3,0}$ can be found from $N_{2,0}$ and $N_{2,1}$. If the derived value of $N_{3,0}$ matches the value in the signed root, the user is assured that L_2 is valid (assuming that the hash function and digital signature are secure).

Confirmations do not need to come from a trusted source because they are self-verifying; defective or expired confirmations will be rejected. Locations for Confirmation Issuers can thus be chosen for their network connectivity.

Users can even provide their own confirmations with their certificates. For example, senders of e-mail messages can include their own confirmations so that

recipients need not fetch them separately. For receive-only wireless devices, piggybacked confirmations are essential.

The CRT architecture meets the technical requirements for most applications. The cryptographic security is provably as good as the underlying hash and signature functions. The only digital signing process (for the tree root) is performed off-line, so signing keys do not need to be managed on computers connected to the network. CRTs issuance is also auditable, since the process of repackaging revocation information into a CRT is predictable and verifiable using the CRT issuer's public key. Performance is excellent, since confirmation issuance requires almost no computational effort and only one signature verification is required to validate a certificate chain. For reliability, multiple Confirmation Issuers can be deployed, so any single failure will not cause an outage.

Communication bandwidth is small and extremely scalable, since the size of a confirmation only increases as the logarithm (base 2) of the number of leaves in the tree. (For example, in a tree with one trillion leaves, there will be 40 connecting hashes, requiring 800 bytes of hashes using 20-byte SHA hashes. With a 1028-bit (128-byte) RSA signature, the confirmation is still less than one kilobyte.) Because confirmations are small and the Confirmation Issuers can be placed close to the users (rather than at the CAs), network latency is minimized. For example, ValiCert's public Confirmation Issuers are placed on major Internet backbones. When an acceptable confirmation is piggybacked with a certificate, no extra network round trips are required.

While CRT systems are well suited to large infrastructures, they can be used in smaller environments as well. In a corporate network, for example, an internal confirmation issuer can provide confirmations for both internal certificates (for which revocation information is not globally distributed) as well as external certificates. In such cases, separate CRTs are typically used.

7 The Future

The need for fast, efficient revocation is not yet clearly understood by many customers, but will become crucial as public key systems are increasingly used to protect valuable assets and attackers learn to profit from compromised keys. Revocation checking will inevitably become a standard, required step whenever a certificate or digital signature is accepted.

References

1. "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework," ITU-T Recommendation X.509 (1197 E), June 1997.
2. P. Kocher and A. Malpani, "Certificate Revocation Trees," ValiCert Inc. Technical Specification, `http://www.valicert.com`.
3. R. Merkle, "Secrecy, Authentication, and Public Key Systems," Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, 1979.
4. National Institute of Standards and Technology, "Secure Hash Standard," Federal Information Processing Standards Publication 180-1, April 1995.