Privacy-Preserving Revocation Checking with Modified CRLs

Maithili Narasimha and Gene Tsudik

Computer Science Department
University of California, Irvine
{mnarasim,gts}@ics.uci.edu

Abstract. Certificate Revocation Lists (CRLs) are a popular means of revocation checking. A CRL is a signed and time-stamped list containing information about all revoked certificates issued by a certification authority. One of the shortcomings of CRLs is poor scalability, which influences update, bandwidth and storage costs. We claim that other (more efficient) revocation techniques leak potentially sensitive information. Information leaks occur since third parties (agents, servers) of dubious trustworthiness discover the identities of the parties posing revocation check queries as well as identities of the queries' targets. An even more important privacy loss results from the third party's ability to tie the source of the revocation check with the query's target. (Since, most likely, the two are about to communicate.) This paper focuses on privacy and efficiency in revocation checking. Its main contribution is a simple modified CRL structure that allows for efficient revocation checking with customizable levels of privacy.

Keywords: Anonymity and Privacy, Certificate Revocation.

1 Introduction and Motivation

Public key cryptography allows entities to establish secure communication channels without pre-established shared secrets. While entities can be assured that communication is confidential, there is no guarantee of authenticity. Authenticity is obtained by binding a public key to some claimed identity or name which is later verified via digital signatures in conjunction with public key certificates (PKCs). A public key certificate, signed by a recognized certification authority (CA), is used to verify the validity, authenticity and ownership of a public key. As long as the issuing CA is trusted, anyone can verify the CA's certificate signature and bind the included name/identity to the public key. Public key certificates work best in large interconnected open systems, where it is generally infeasible to directly authenticate the owners of all public keys. X.509 [23] is one well-known certificate format widely used in several Internet-related contexts. The peer-based PGP/GPG [2,7] format represents another popular approach.

Since a certificate is a form of a capability, one of the biggest problems associated with large-scale use of certificates is *revocation*. There are many reasons

J. Lopez, P. Samarati, and J.L. Ferrer (Eds.): EuroPKI 2007, LNCS 4582, pp. 18–33, 2007.

that can lead to a certificate being revoked prematurely. They include [23]: loss or compromise of a private key, change of affiliation or job function, algorithm compromise, or change in security policy. To cope with revocation, it must be possible to check the status of any certificate at any time.

Revocation techniques can be roughly partitioned into implicit and explicit classes. In the former, each certificate owner possesses a timely proof of non-revocation which it supplies on demand to anyone. Lack of such a proof implicitly signifies revocation. An example of implicit revocation is the Certificate Revocation System (CRS) [17]. Most revocation methods are explicit, i.e., they involve generation, maintenance and distribution of various secure data structures that contain revocation information for a given CA or a given range of certificates.

Certificate Revocation Lists (CRLs) represent the most widely used means of explicit revocation checking. Each certificate issuer periodically generates a signed list of revoked certificates and publishes it at (usually untrusted) public directories or servers. Inclusion of a certificate in the list signifies explicit revocation. Verifiers retrieve and cache the latest CRL and use it during certificate validation. Typically, a revoked certificate is included in a CRL from the time it is revoked until its validity period expires. Since certificate lifetime is typically measured in years, even modest revocation rates can result in very long accumulated CRLs. In bandwidth-constrained environments, transferring such CRLs can be expensive. Furthermore, since CRLs are published periodically, another potential concern is that many verifiers may request them around the time of publication. The burst of requests immediately following CRL publication may result in very high network traffic and can cause congestion. Thus, one of the biggest disadvantages of CRLs is the high cost associated with updating and querying the lists and this raises serious scalability concerns.

Other well-known explicit revocation methods include Certificate Revocation Trees (CRTs) [12] and Skip-Lists [8]. Another prominent technique is the On-line Certificate Status Protocol (OCSP) [18] which involves a multitude of validation agents (VAs) which respond to client queries with signed replies indicating current status of a target certificate. However, these explicit revocation methods have an unpleasant side-effect: they divulge too much information. Specifically, a third party (agent, server, responder or distribution point) of dubious trustworthiness knows: (1) the entity requesting the revocation check (source), and (2) the entity whose status is being checked (target). An even more important **loss of privacy** results from the third party tying the source of the revocation checking query to that query's target. This is significant, because revocation status check typically serves as a prelude to actual communication between the two parties. (We assume that communication between verifiers and on-line revocation agents (third parties) is private, i.e., conducted over secure channels protected by tools such as IPsec [10] or SSL/TLS [9,6].)

Given the continual assault on privacy by governments, spammers and just plain hackers, privacy leakage in certificate revocation checking is an important issue worth considering. Consider, for example, a certain country with a