Network Working Group
Request for Comments: 2510
Category: Standards Track

C. Adams
Entrust Technologies
S. Farrell
SSE

March 1999

Internet X.509 Public Key Infrastructure Certificate Management Protocols

Status of this Memo

Copyright Notice

Abstract

This document describes the Internet X.509 Public Key Infrastructure (PKI) Certificate Management Protocols. Protocol messages are defined for all relevant aspects of certificate creation and management. Note that "certificate" in this document refers to an X.509v3 Certificate as defined in [COR95, X509-AM].

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document (in uppercase, as shown) are to be interpreted as described in [RFC2119].

Introduction

The layout of this document is as follows:

- Section 1 contains an overview of PKI management;
- Section 2 contains discussion of assumptions and restrictions;
- Section 3 contains data structures used for PKI management messages;
- Section 4 defines the functions that are to be carried out in PKI management by conforming implementations;
- Section 5 describes a simple protocol for transporting PKI messages;
- the Appendices specify profiles for conforming implementations and provide an ASN.1 module containing the syntax for all messages defined in this specification.

1 PKI Management Overview

The PKI must be structured to be consistent with the types of individuals who must administer it. Providing such administrators with unbounded choices not only complicates the software required but also increases the chances that a subtle mistake by an administrator or software developer will result in broader compromise. Similarly, restricting administrators with cumbersome mechanisms will cause them not to use the PKI.

Management protocols are REQUIRED to support on-line interactions between Public Key Infrastructure (PKI) components. For example, a management protocol might be used between a Certification Authority (CA) and a client system with which a key pair is associated, or between two CAs that issue cross-certificates for each other.

1.1 PKI Management Model

Before specifying particular message formats and procedures we first define the entities involved in PKI management and their interactions (in terms of the PKI management functions required). We then group these functions in order to accommodate different identifiable types of end entities.

1.2 Definitions of PKI Entities

The entities involved in PKI management include the end entity (i.e., the entity to be named in the subject field of a certificate) and the certification authority (i.e., the entity named in the issuer field of a certificate). A registration authority MAY also be involved in PKI management.

1.2.1 Subjects and End Entities

The term "subject" is used here to refer to the entity named in the subject field of a certificate; when we wish to distinguish the tools and/or software used by the subject (e.g., a local certificate management module) we will use the term "subject equipment". In general, the term "end entity" (EE) rather than subject is preferred in order to avoid confusion with the field name.

It is important to note that the end entities here will include not only human users of applications, but also applications themselves (e.g., for IP security). This factor influences the protocols which the PKI management operations use; for example, application software is far more likely to know exactly which certificate extensions are required than are human users. PKI management entities are also end entities in the sense that they are sometimes named in the subject

field of a certificate or cross-certificate. Where appropriate, the term "end-entity" will be used to refer to end entities who are not PKI management entities.

All end entities require secure local access to some information -- at a minimum, their own name and private key, the name of a CA which is directly trusted by this entity and that CA's public key (or a fingerprint of the public key where a self-certified version is available elsewhere). Implementations MAY use secure local storage for more than this minimum (e.g., the end entity's own certificate or application-specific information). The form of storage will also vary -- from files to tamper-resistant cryptographic tokens. Such local trusted storage is referred to here as the end entity's Personal Security Environment (PSE).

Though PSE formats are beyond the scope of this document (they are very dependent on equipment, et cetera), a generic interchange format for PSEs is defined here - a certification response message MAY be used.

1.2.2 Certification Authority

The certification authority (CA) may or may not actually be a real "third party" from the end entity's point of view. Quite often, the CA will actually belong to the same organization as the end entities it supports.

Again, we use the term CA to refer to the entity named in the issuer field of a certificate; when it is necessary to distinguish the software or hardware tools used by the CA we use the term "CA equipment".

The CA equipment will often include both an "off-line" component and an "on-line" component, with the CA private key only available to the "off-line" component. This is, however, a matter for implementers (though it is also relevant as a policy issue).

We use the term "root CA" to indicate a CA that is directly trusted by an end entity; that is, securely acquiring the value of a root CA public key requires some out-of-band step(s). This term is not meant to imply that a root CA is necessarily at the top of any hierarchy, simply that the CA in question is trusted directly.

A "subordinate CA" is one that is not a root CA for the end entity in question. Often, a subordinate CA will not be a root CA for any entity but this is not mandatory.

1.2.3 Registration Authority

In addition to end-entities and CAs, many environments call for the existence of a Registration Authority (RA) separate from the Certification Authority. The functions which the registration authority may carry out will vary from case to case but MAY include personal authentication, token distribution, revocation reporting, name assignment, key generation, archival of key pairs, et cetera.

This document views the RA as an OPTIONAL component - when it is not present the CA is assumed to be able to carry out the RA's functions so that the PKI management protocols are the same from the end-entity's point of view.

Again, we distinguish, where necessary, between the RA and the tools used (the "RA equipment").

Note that an RA is itself an end entity. We further assume that all RAs are in fact certified end entities and that RAs have private keys that are usable for signing. How a particular CA equipment identifies some end entities as RAs is an implementation issue (i.e., this document specifies no special RA certification operation). We do not mandate that the RA is certified by the CA with which it is interacting at the moment (so one RA may work with more than one CA whilst only being certified once).

In some circumstances end entities will communicate directly with a CA even where an RA is present. For example, for initial registration and/or certification the subject may use its RA, but communicate directly with the CA in order to refresh its certificate.

1.3 PKI Management Requirements

The protocols given here meet the following requirements on PKI management.

1. PKI management must conform to the ISO 9594-8 standard and the associated amendments (certificate extensions)

2. PKI management must conform to the other parts of this series.

3. It must be possible to regularly update any key pair without affecting any other key pair.

4. The use of confidentiality in PKI management protocols must be kept to a minimum in order to ease regulatory problems.

5. PKI management protocols must allow the use of different industry-standard cryptographic algorithms, (specifically including RSA, DSA, MD5, SHA-1) -- this means that any given CA, RA, or end entity may, in principle, use whichever algorithms suit it for its own key pair(s).

6. PKI management protocols must not preclude the generation of key pairs by the end-entity concerned, by an RA, or by a CA -- key generation may also occur elsewhere, but for the purposes of PKI management we can regard key generation as occurring wherever the key is first present at an end entity, RA, or CA.

7. PKI management protocols must support the publication of certificates by the end-entity concerned, by an RA, or by a CA. Different implementations and different environments may choose any of the above approaches.

8. PKI management protocols must support the production of Certificate Revocation Lists (CRLs) by allowing certified end entities to make requests for the revocation of certificates - this must be done in such a way that the denial-of-service attacks which are possible are not made simpler.

9. PKI management protocols must be usable over a variety of "transport" mechanisms, specifically including mail, http, TCP/IP and ftp.

10. Final authority for certification creation rests with the CA; no RA or end-entity equipment can assume that any certificate issued by a CA will contain what was requested -- a CA may alter certificate field values or may add, delete or alter extensions according to its operating policy. In other words, all PKI entities (end-entities, RAs, and CAs) must be capable of handling responses to requests for certificates in which the actual certificate issued is different from that requested (for example, a CA may shorten the validity period requested). Note that policy may dictate that the CA must not publish or otherwise distribute the certificate until the requesting entity has reviewed and accepted the newly-created certificate (typically through use of the PKIConfirm message).

11. A graceful, scheduled change-over from one non-compromised CA key pair to the next (CA key update) must be supported (note that if the CA key is compromised, re-initialization must be performed for all entities in the domain of that CA). An end entity whose PSE contains the new CA public key (following a CA key update) must also be able to verify certificates verifiable using the old public key. End entities who directly

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.