TOC

Network Working Group	R. Reddy
Internet-Draft	National Security Agency
Intended status: Informational	C. Wallace
Expires: April 6, 2009	Cygnacom Solutions
	October 03, 2008

Trust Anchor Management Requirements
draft-ietf-pkix-ta-mgmt-reqs-01

Status of this Memo

Abstract

A trust anchor represents an authoritative entity via a public key and associated data. The public key is used to verify digital signatures and the associated data is used to constrain the types of information for which the trust anchor is authoritative. A relying party uses trust anchors to determine if a digitally signed object is valid by verifying a digital signature using the trust anchor's public key, and by enforcing the constraints expressed in the associated data for the trust anchor. This document describes some of the problems associated with the lack of a standard trust anchor management mechanism and defines requirements for data formats and protocols designed to address these problems. This document discusses only public keys as trust anchors; symmetric key trust anchors are not considered.

Table of Contents

1. Introduction

TOC

Digital signatures are used in many applications. For digital signatures to provide integrity and authentication, the public key used to verify the digital signature must be "trusted", i.e., accepted by a relying party (RP) as appropriate for use in the given context. A public key used to verify a signature must be configured as a trust anchor or contained in a certificate that can be transitively verified by a certification path terminating at a trust anchor. A Trust Anchor is a public key and associated data used by a relying party to validate a signature on a signed object where the object is either:

*a public key certificate that begins a certification path terminated by a signature certificate or encryption certificate

*an object (other than a public key certificate) that cannot be validated via use of a certification path

Trust anchors have only local significance, i.e., each RP is configured with a set of trust anchors, either by the RP or by an entity that manages TAs in the context in which the RP operates. The associated data defines the scope of a trust anchor by imposing constraints on the signatures the trust anchor may be used to verify. For example, if a trust anchor is used to verify signatures on X.509 certificates, these constraints may include a combination of name spaces, certificate policies, or application/usage types.

One use of digital signatures is the verification of signatures on firmware packages loaded into hardware modules, such as cryptographic modules, cable boxes, routers, etc. Since such devices are often managed remotely, the devices must be able to authenticate the source of management interactions and can use trust anchors to perform this authentication. However, trust anchors require management as well.

All applications that rely upon digital signatures rely upon some means of managing one or more sets of trust anchors. These sets of trust anchors are referred to in this document as trust anchor stores. Often, the means of managing trust anchor stores are application-specific and rely upon out-of-band means to establish and maintain trustworthiness. An application may use multiple trust anchor stores and a given trust anchor store may be used by multiple applications. Trust anchor stores are managed by trust anchor managers.

This section provides an introduction and defines basic terminology. Section 2 describes problems with current trust anchor management methods. Sections 3 and 4 describe requirements and security considerations for a trust anchor management solution.

1.1. Terminology

TOC

The following terms are defined in order to provide a vocabulary for describing requirements for trust anchor management.

> **Trust Anchor:** A trust anchor represents an authoritative entity via a public key and associated data. The public key is used to verify digital signatures and the associated data is used to constrain the types of information for which the trust anchor is authoritative. A relying party uses trust anchors to determine if a digitally signed object is valid by verifying a digital signature using the trust anchor's public key, and by enforcing the constraints expressed in the associated data for the trust anchor.
>
> **Trust Anchor Manager:** Trust anchor manager is a role responsible for managing the contents of a trust anchor store. Throughout this document, trust anchor managers are assumed to be represented as trust anchors.
>
> **Trust Anchor Store:** A trust anchor store is a set of one or more trust anchors. A trust anchor store may be managed by one or more trust anchor managers. A device may have more than one trust anchor store.

2. Problem Statement

TOC

Trust anchors are used to support many application scenarios. Most Internet browsers and email clients use trust anchors when authenticating TLS sessions, verifying signed email and generating encrypted email by validating a certification path to a server's certificate, an e-mail originator's certificate or an e-mail recipient's certificate. Many software distributions are digitally signed to enable authentication of the software source to be performed prior to installation. Trust anchors that support these applications are typically installed as part of the operating system (OS) or application, installed using an enterprise configuration management system or installed directly by an OS or application user.

Trust anchors are typically stored in application-specific or operating system-specific trust anchor stores. Often, a single machine may have a number of different trust anchor stores that may not be synchronized. Reviewing the contents of a particular trust anchor store typically involves use of a proprietary tool that interacts with a particular type of trust store.

The presence of a trust anchor in a particular store often conveys implicit authorization to validate signatures for any contexts from which the store is accessed. For example, the public key of a timestamp authority (TSA) may be installed in a trust anchor store to validate signatures on timestamps. However, if the store containing this TA is used by multiple applications that serve different purposes, the same key may be used (inappropriately) to validate other types of objects such as certificates or OCSP responses. There is currently no standard means of limiting the applicability (scope) of a trust anchor except by placing different TAs in different stores and limiting the set of applications that access a given TA store.

Trust relationships between PKIs are negotiated by policy authorities. Negotiations frequently require significant time to ensure all participating parties' requirements are satisfied. These requirements are expressed, to some extent, in public key certificates via policy constraints, name constraints and etc. In order for these requirements to be enforced, trust anchor stores must be managed in accord with policy authority intentions and avoid circumventing constraints defined in a cross-certificate by recognizing the subject of the cross certificate as a trust anchor.

Trust anchors are often represented as self-signed certificates, which provide no useful means of establishing the validity of the information contained in the certificate. Confidence in the integrity of a trust anchor is typically established through out-of-band means, often by checking the "fingerprint" (one-way hash) of the self-signed certificate with an authoritative source. Routine trust anchor re-key operations typically require similar out-of-band checks. Ideally, only the initial set of trust anchors installed in a particular trust anchor store should require out-of-band verification, particularly when the costs of performing out-of-band checks commensurate with the security requirements of applications using the trust anchor store are high.

Despite the prevalent use of trust anchors, there is neither a standard means for discovering which trust anchors installed in a particular trust anchor store nor a standard means of managing those trust anchors. The remainder of this document describes requirements for a solution to this problem along with some security considerations.

TOC

3. Requirements

This section describes the requirements for a trust anchor management protocol. Requirements are provided for trust anchor contents as well as for trust anchor store management operations.

3.1. Transport independence

TOC

3.1.1. Functional Requirements

TOC

A general-purpose solution for the management of trust anchors must be transport independent in order to apply to a range of device communications environments. It should be applicable in both session-oriented and store-and-forward contexts. Message integrity must be assured in all cases.

3.1.2. Rationale

TOC

Not all devices that use trust anchors are available for online management operations; some devices may require manual interaction for trust anchor management. Message integrity is required to authenticate the originator of a TA management transaction and confirm the authorization of the originator for that transaction.

3.2. Basic management operations

TOC

3.2.1. Functional Requirements

TOC

At a minimum, a protocol used for trust anchor management must enable a trust anchor manager to perform the following operations:

- *Determine which trust anchors are installed in a particular trust anchor store

*Add one or more trust anchors to a trust anchor store

*Remove one or more trust anchors from a trust anchor store

*Replace an entire trust anchor store

A trust anchor management protocol must provide support for these basic operations, however, not all implementations must support each option. For example, some implementations may only support replacement of trust anchor stores.

3.2.2. Rationale

TOC

These requirements describe the core operations required to manage the contents of a trust anchor store. An edit operation was omitted for sake of simplicity, with consecutive remove and add operations used for this purpose. Add and remove operations are compound to avoid unnecessary round trips and are provided to avoid always replacing an entire trust anchor store. Trust anchor store replacement may be useful as a simple, higher bandwidth alternative to add and remove operations. Many devices and some applications utilize multiple trust anchor stores.

3.3. Management targets

TOC

3.3.1. Functional Requirements

TOC

A protocol for TA management must allow a TA management transaction to be directed to:

> All TA stores for which the manager is responsible
>
> An enumerated list of one or more groups of trust anchor stores
>
> An individual trust anchor store

TOC

3.3.2. Rationale

Trust anchor configurations may be uniform across an enterprise, or they may be unique to a single application or small set of applications.

Connections between PKIs can be accomplished using different means. Unilateral or bilateral cross-certification can be performed, or a community may simply elect to explicitly accept a trust anchor from another community. Typically, these decisions occur at the enterprise level. In some scenarios, it can be useful to establish these connections for a small community within an enterprise. Enterprise-wide mechanisms such as cross-certificates are ill-suited for this purpose since certificate revocation or expiration affects the entire enterprise. A trust anchor management protocol can address this issue by supporting limited installation of trust anchors and by supporting expression of constraints on trust anchor usage. Limited installation requires the ability to identify the members of the community that are authorized to rely upon a particular trust anchor, as well as the ability to query and report on the contents of trust anchor stores. The trust anchor constraints can represent the limitations that would have been expressed in a cross-certificate and limited installation ensures the recognition of the trust anchor does not necessarily encompass an entire enterprise.

3.4. Delegation of TA Management Authority

TOC

3.4.1. Functional Requirements

TOC

A trust anchor management protocol must enable secure transfer of control of a trust anchor store from one trust anchor manager to another. It must also enable delegation for specific operations without requiring delegation of the overall trust anchor management capability itself.

3.4.2. Rationale

TOC

Trust anchor re-key is one type of transfer that must be supported. In this case, the new key will be assigned the same privileges as the old key. Creation of trust anchors for specific purposes, such as firmware signing, is another example of delegation. For example, a trust anchor manager may delegate only the authority to sign firmware and disallow further delegation of the privilege, or the trust anchor manager may allow its delegate to delegate firmware signing to other entities.

3.5. RFC 5280 Support

TOC

3.5.1. Functional Requirements

TOC

A trust anchor management protocol must enable management of trust anchors that can be used to validate certification paths in accordance with [RFC5280] (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008.) and [RFC5055] (Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)," December 2007.). A trust anchor format must enable the representation of constraints that influence certification path validation or otherwise establish the scope of usage of the trust anchor public key. Examples of such constraints are name constraints, certificate policies and key usage.

3.5.2. Rationale

TOC

Certification path validation is one of the most common applications of trust anchors. The rules for using trust anchors for path validation are established in [RFC5280] (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," May 2008.). [RFC5055] (Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)," December 2007.) describes the use of trust anchors for delegated path validation.

3.6. Support Purposes Other Than Certification Path Validation

TOC

3.6.1. Functional Requirements

TOC

A trust anchor management protocol must enable management of trust anchors that can be used for purposes other than certification path validation, including trust anchors that cannot be used for certification path validation. It should be possible to authorize a trust anchor to delegate authority (to other TAs or certificate holders) and to prevent a trust anchor from delegating authority.

3.6.2. Rationale

TOC

Trust anchors are used to validate a variety of objects other than public key certificates and CRLs. For example, a trust anchor may be used to verify firmware packages [RFC4108] (Housley, R., “Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages,” August 2005.), OCSP responses [RFC2560] (Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” June 1999.), SCVP responses [RFC5055] (Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, “Server-Based Certificate Validation Protocol (SCVP),” December 2007.) or timestamps [RFC3161] (Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, “Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP),” August 2001.). TAs authorized for these operations may not be authorized to sign public key certificates or CRLs.

3.7. Trust Anchor Format

TOC

3.7.1. Functional Requirements

TOC

Minimally, a trust anchor management protocol must support management of trust anchors represented as self-signed certificates and trust anchors represented as a distinguished name and public key information. The definition of a trust anchor must include a public key, a public key algorithm and, if necessary, public key parameters. When the public key is used to validate certification paths, a distinguished name also must be included per [RFC3852] (Housley, R., “Cryptographic Message Syntax (CMS),” July 2004.). A trust anchor format should enable specification of public key identifier to enable other applications of the trust anchor, for example, verification of data signed using the Cryptographic Message Syntax (CMS) SignedData structure [RFC3852] (Housley, R., “Cryptographic Message Syntax (CMS),” July 2004.). A trust anchor format should also enable the use of constraints that can be applied to specify the type/usage of a trust anchor.

3.7.2. Rationale

TOC

There is no standardized format for trust anchors. Self-signed X.509 certificates are typically used but [RFC5280] (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” May 2008.) does not mandate a particular trust anchor representation. It requires only that a trust anchor's public key information and distinguished name be available during certification path validation. CMS is widely used to protect a variety of types of content using digital signatures, including contents that may verified directly using a trust anchor, such as firmware packages [RFC4108] (Housley, R., “Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages,” August 2005.).

3.8. Authentication of Trust Anchor Store Contents

TOC

3.8.1. Functional Requirements

TOC

A trust anchor manager must be able to authenticate which trust anchor store corresponds to a report listing the contents of the trust anchor store and be able to confirm the contents of the report have not been subsequently altered. Replay of old reports (from the same trust anchor store) must be detectable by a TA manager.

3.8.2. Rationale

TOC

Authentication of trust anchor store reports is required to support remote management operations.

3.9. Source Authentication

TOC

3.9.1. Functional Requirements

TOC

An entity receiving trust anchor management data must be able to authenticate the party providing the information and must be able to confirm the party is authorized to provide that trust anchor information.

3.9.2. Rationale

TOC

A trust anchor manager may be authorized to participate in trust anchor management protocol exchanges, but be limited to managing trust anchors within a particular scope. Alternatively, a trust anchor manager may be authorized to participate in trust anchor management protocol exchanges without any constraints on the types of trust anchors that may be managed.

3.10. Reduce Reliance on Out-of-Band Trust Mechanisms

TOC

3.10.1. Functional Requirements

TOC

A trust anchor management protocol should enable TA integrity to be checked automatically without relying on out-of-band trust mechanisms.

3.10.2. Rationale

TOC

Traditionally, a trust anchor is distributed out-of-band with its integrity checked manually prior to installation. Installation typically is performed by anyone with sufficient administrative privilege on the system receiving the trust anchor. Reliance on out-of-band trust mechanisms is one problem with current trust anchor management approaches and reduction of the need to use out-of-band trust mechanisms is a primary motivation for developing a trust anchor management protocol. Ideally, out-of-band trust mechanisms will be required only during trust anchor store initialization.

3.11. Replay Detection

TOC

3.11.1. Functional Requirements

TOC

A trust anchor management protocol must enable participants engaged in a trust anchor management protocol exchange to detect replay attacks. Replay detection mechanisms should not introduce a requirement for a reliable source of time as some devices that utilize trust anchors have no access to a reliable source of time.

3.11.2. Rationale

TOC

Replay of old trust anchor management messages could result in the addition of compromised trust anchors to a trust anchor store, potentially exposing applications to malicious signed objects or certification paths.

3.12. Compromise or Disaster Recovery

TOC

3.12.1. Functional Requirements

TOC

A trust anchor management protocol must enable recovery from the compromise or loss of a trust anchor private key, including the private key authorized to serve as a source of trust anchor information.

TOC

3.12.2. Rationale

Compromise or loss of a private key corresponding to a trust anchor can have significant negative consequences. Currently, in some cases, re-initialization of all effected trust anchor stores is required to recover from a lost or compromised trust anchor key. A trust anchor management protocol should support recovery options that do not require trust anchor store re-initialization.

3.13. Usage of Trust Anchor Information for Certification Path Validation

TOC

3.13.1. Functional Requirements

TOC

RFC5280 requires subject name and public key and leaves the usage of other information, such as name constraints extensions, as optional. Where a trust anchor management protocol is used, constraints must be observed if included in a trust anchor.

3.13.2. Rationale

TOC

Inclusion of constraints in trust anchor objects is optional. Where constraints are established by a trust anchor manager using a trust anchor management protocol, there must exist an expectation of enforcement to ensure consistent behavior across applications. Legacy considerations prevent requiring enforcement in all cases where a trust anchor is used.

---

4. Security Considerations

TOC

The public key used to authenticate a TA management transaction may have been placed in the client as the result of an earlier TA management transaction or during an initial bootstrap configuration operation. In most scenarios, at least one public key authorized for trust anchor management must be placed in each trust anchor store to be managed during the initial configuration of the trust anchor store. This public key may be transported and checked using out-of-band means. In all scenarios, regardless of the authentication mechanism, at least one trust anchor manager must be established for each trust anchor store during the initial configuration of the trust anchor store. Many of the security considerations from [RFC5280] (Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” May 2008.) are also applicable to trust anchor management.

5. IANA Considerations

TOC

None. Please remove this section prior to publication as an RFC.

6. References

TOC

6.1. Normative References

TOC

[RFC5055]	Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, “Server-Based Certificate Validation Protocol (SCVP),” RFC 5055, December 2007 (TXT).
[RFC5280]	Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC 5280, May 2008 (TXT).

6.2. Informative References

TOC

[RFC2560]	Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” RFC 2560, June 1999 (TXT).
[RFC3161]	Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, “Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP),” RFC 3161, August 2001 (TXT).
[RFC3852]	Housley, R., “Cryptographic Message Syntax (CMS),” RFC 3852, July 2004 (TXT).
[RFC4108]	Housley, R., “Using Cryptographic Message Syntax (CMS) to Protect Firmware Packages,” RFC 4108, August 2005 (TXT).

Authors' Addresses

TOC

Raksha Reddy
National Security Agency
Suite 6599
9800 Savage Road
Fort Meade, MD 20755
Email: r.reddy@radium.ncsc.mil

Carl Wallace
Cygnacom Solutions
Suite 5200
7925 Jones Branch Drive
McLean, VA 22102
Email: cwallace@cygnacom.com

Full Copyright Statement

TOC

Intellectual Property