Implementation GUIDELINE

AKID/SKID Implementation Guideline

September 2002

The Subject Key Identifier (SKID) is used to help construct certification paths. Since there are multiple methods for calculating the SKID, care must be exercised to ensure that the SKID populated in self-signed certificate of a Certification Authority (CA) matches the SKID populated in any cross-certificates issued for that CA by other CAs. The purpose of this implementation guideline is to recommend how the SKID should be populated in cross-certificates.

Acknowledgements

Implementation Guideline is a deliverable from the PKI Forum's Technical Working Group (TWG). Several member organizations and individuals have contributed by providing content, editorial assistance and editorial reviews.

Author:

Steve Lloyd
PKI Forum

Background

The Authority Key Identifier (AKID) and Subject Key Identifier (SKID) are certificate extensions that can be used to help facilitate the certification path construction process. As discussed in X.509 and the Internet Certificate and CRL Profile (RFC3280), AKIDs are used to distinguish one public key from another when a given Certification Authority (CA) has multiple signing keys, and SKIDs provide a means to identify certificates that contain a specific public key. Similar to "name chaining" between a trust anchor and an end-entity certificate (where the DN of the subject in the first certificate is the DN of the issuer in the next certificate in the path, and so on), the SKID of the first certificate should be the AKID of the next certificate in the path, and so on.

The AKID can be represented using a *keyIdentifier*, or a *authorityCertIssuer*|*authoritySerialNumber* pair, or both. However, the Internet Certificate and CRL Profile states that the keyIdentifier field in the AKID must be included in all certificates generated by conforming CAs (with the exception of self-signed certificates, in which case the *keyIdentifier* would be included within the SKID and, optionally, within the AKID). More importantly, the SKID syntax is defined in terms of the *keyIdentifier* only, and the Internet Certificate and CRL Profile states that all CA certificates must include the SKID extension (which contains the keyIdentifier for the public key contained within that certificate). As specified in the Internet Certificate and CRL Profile, there are multiple methods available for calculating the *keyIdentifier*. While one of these methods should be used, the only mandatory requirement is that the value of the keyIdentifier be unique (within the context of a given CA) for a given public key. See the PKI Forum Certification Path Construction white paper available from http://www.pkiforum.org/resources.html for more information regarding the keyIdentifier.

When two otherwise isolated PKI domains wish to interoperate, a CA in one domain cross-certifies with a CA in the other domain, and vice versa. This is referred to as bilateral or mutual cross-certification. During the mutual cross-certification process, each CA issues a cross-certificate to the other CA. Essentially, this means that the public key of one CA is signed by the other CA and vice versa.

Cross-certification can also be unilateral. This is typically found in hierarchical trust models where superior CAs issue certificates to subordinate CAs. In this case there is only one cross-certificate (i.e., the superior CA issues a cross-certificate to the subordinate, but not vice versa).

Contents

PKI forum

Problem Statement

When the CA requesting a cross-certificate already has a self-signed certificate, the *keyIdentifier* associated with that particular public key will be populated in the SKID extension of that self-signed certificate. If the issuing CA selects a different method for calculating the keyIdentifier associated with the subject CA, then a mismatch between the *keyIdentifier* in the SKID of the self-signed certificate and the keyIdentifier in the SKID in the cross-certificate will occur. Both the U.S. Federal Bridge CA and CESG interoperability initiatives demonstrated that these mismatches did occur and that this prevented proper certification path construction.

A method is required to ensure that the keyIdentifer populated in the SKID extension of a cross-certificate is the exact same value that is populated in the SKID extension of the corresponding self-signed certificate.

Therefore, when one CA cross-certifies with another, a method is required to ensure that the keyIdentifier populated in the SKID extension of a cross-certificate is the exact same value that is populated in the SKID extension of the corresponding self-signed certificate.[1]

Options Analysis

Three options that might be employed to solve this problem were explored. The first option is a non-starter, the second is doable but is a bit wasteful, and the third seems to be the simplest and most straightforward to implement. Each option is discussed briefly below.

The first option considered was to mandate a single algorithm for calculating the *keyIdentifier*. However, since the applicable standards allow for more than one algorithm, these standards would have to be changed (assuming that we could reach agreement on a single algorithm - which is a bit presumptuous in itself). Further, numerous CAs have been deployed that would not conform to the selected algorithm (regardless of which algorithm was selected). This would render those CAs non-conformant and would require retrofitting of existing products. Given these difficulties, this option is considered to be problematic.

The second option considered was to establish Object Identifiers for all possible algorithms that could be used to calculate the *keyIdentifier*, and to convey the algorithm to be used to the issuing CA as part of the cross-certificate request. However, this would require additional standards work, and it seems wasteful to recalculate something that is already known.

Since the key Identifier has already been calculated and populated in the self-signed CA certificate, it seems reasonable to supply this information to the issuing CA as part of the cross-certificate request.

This leads us to the third option. Since the *keyIdentifier* has already been calculated and populated in the self-signed CA certificate, it seems reasonable to supply this information to the issuing CA as part of the cross-certificate request. Existing protocols have placeholders for conveying this information, as illustrated in Annex A.

[1] In a strict hierarchy, subordinate CAs do not typically have self-signed certificates, and the interoperability problem associated with the keyIdentifier would not be an issue. However, there are circumstances where subordinate CAs do have self-signed certificates even though they belong to what is traditionally associated as a hierarchy. In addition, there may be other reasons that a subordinate CA may want to specify a particular *keyIdentifier*. Thus, we allow for the case where a CA initiating a cross-certification with a superior can supply a specific *keyIdentifier*.

Endorsement

In order to eliminate the potential for SKID mismatches during the cross-certification process, the PKI Forum endorses the third option described above. In particular, the PKI Forum agrees with the following:

> The keyIdentifier populated in the SKID extension of the requesting CA's applicable self-signed certificate MUST be conveyed to the issuing CA within the cross-certificate request, and the issuing CA MUST populate the *keyIdentifier* in the SKID of the associated cross-certificate with this value. This may be accomplished via out-of-band means in the case of off-line cross-certification, or it may be accomplished via protocol in the case of on-line cross-certification. If the *keyIdentifier* is not present, it is assumed that the cross-certification pertains to a subordinate CA that does not have a self-signed certificate, and the issuing CA is free to calculate the *keyIdentifier* using one of the methods defined in RFC3280.

Annex A: Specific Protocol Usage Examples

The more common cross-certification protocols for on-line cross-certification are based on PKCS #10 as defined in RFC 2986 or on the Certificate Request Message Format (CRMF) as defined in RFC2511. In order to help ensure interoperability among different vendors, the following examples are provided:

PKCS #10:

The PKCS #10 Certification Request message can be used to request end-user certificates as well as cross-certificates. The Certification Request message includes an attributes element which, among other things, is a placeholder for certificate extensions as defined in PKCS #9 (RFC2985) and X.509. Therefore, it is possible to include the SKID extension/value within the PKCS #10 Certification Request message. CAs with self-signed certificates must include the same SKID value in the SKID extension of the Certification Request message in accordance with PKCS #10, PKCS #9 and X.509.

CRMF:

CRMF defines a Certificate Request Message. In this case, the Certificate Request Message includes a certificate template attribute that allows the requestor to specify as much of the certificate content as it desires. The certificate template syntax includes extensions which can be specified in accordance with X.509. Therefore, when a CA that has a self-signed certificate makes a cross-certificate request to another CA using the CRMF Certificate Request Message, the SKID extension/value must be included as one of the certificate extensions conveyed in the certificate template attribute.

About PKI Forum

The PKI Forum is an international, not-for-profit, multi-vendor and end-user alliance whose purpose is to accelerate the adoption and use of Public-Key Infrastructure (PKI).

The PKI Forum advocates cooperation and market awareness to enable organizations to understand and exploit the value of PKI in applications relevant to their business.

Phone: +1 781 876 8810
e-Mail: info@pkiforum.org
Web: www.pkiforum.org