Network Working Group
Request for Comments: 2528
Category: Informational

R. Housley
SPYRUS
W. Polk
NIST

March 1999

Internet X.509 Public Key Infrastructure

Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates

Status of this Memo

Copyright Notice

Table of Contents

Abstract

The Key Exchange Algorithm (KEA) is a classified algorithm for exchanging keys. This specification profiles the format and semantics of fields in X.509 V3 certificates containing KEA keys. The specification addresses the subjectPublicKeyInfo field and the keyUsage extension.

1. Executive Summary

This specification contains guidance on the use of the Internet Public Key Infrastructure certificates to convey Key Exchange Algorithm (KEA) keys. This specification is an addendum to RFC 2459, "Internet X.509 Public Key Infrastructure: Certificate and CRL Profile". Implementations of this specification must also conform to RFC 2459. Implementations of this specification are not required to conform to other parts from that series.

2. Requirements and Assumptions

The goal is to augment the X.509 certificate profile presented in Part 1 to facilitate the management of KEA keys for those communities which use this algorithm.

2.1. Communication and Topology

This profile, as presented in [RFC 2459] and augmented by this specification, supports users without high bandwidth, real-time IP connectivity, or high connection availability. In addition, the profile allows for the presence of firewall or other filtered communication.

This profile does not assume the deployment of an X.500 Directory system. The profile does not prohibit the use of an X.500 Directory, but other means of distributing certificates and certificate revocation lists (CRLs) are supported.

2.2. Acceptability Criteria

The goal of the Internet Public Key Infrastructure (PKI) is to meet the needs of deterministic, automated identification, authentication, access control, and authorization functions. Support for these services determines the attributes contained in the certificate as well as the ancillary control information in the certificate such as policy data and certification path constraints.

The goal of this document is to profile KEA certificates, specifying the contents and semantics of attributes which were not fully specified by [RFC 2459]. If not specifically addressed by this document, the contents and semantics of the fields and extensions must be as described in [RFC 2459].

2.3. User Expectations

Users of the Internet PKI are people and processes who use client software and are the subjects named in certificates. These uses include readers and writers of electronic mail, the clients for WWW browsers, WWW servers, and the key manager for IPSEC within a router. This profile recognizes the limitations of the platforms these users employ and the sophistication/attentiveness of the users themselves. This manifests itself in minimal user configuration responsibility (e.g., root keys, rules), explicit platform usage constraints within the certificate, certification path constraints which shield the user from many malicious actions, and applications which sensibly automate validation functions.

2.4. Administrator Expectations

As with users, the Internet PKI profile is structured to support the individuals who generally operate Certification Authorities (CAs). Providing administrators with unbounded choices increases the chances that a subtle CA administrator mistake will result in broad compromise or unnecessarily limit interoperability. This profile defines the object identifiers and data formats that must be supported to interpret KEA public keys.

3. KEA Algorithm Support

This section describes object identifiers and data formats which may be used with [RFC 2459] to describe X.509 certificates containing a KEA public key. Conforming CAs are required to use the object identifiers and data formats when issuing KEA certificates. Conforming applications shall recognize the object identifiers and process the data formats when processing such certificates.

3.1. Subject Public Key Info

The certificate identifies the KEA algorithm, conveys optional parameters, and specifies the KEA public key in the subjectPublicKeyInfo field. The subjectPublicKeyInfo field is a SEQUENCE of an algorithm identifier and the subjectPublicKey field.

The certificate indicates the algorithm through an algorithm identifier. This algorithm identifier consists of an object identifier (OID) and optional associated parameters. Section 3.1.1 identifies the preferred OID and parameters for the KEA algorithm. Conforming CAs shall use the identified OID when issuing certificates containing public keys for the KEA algorithm. Conforming applications supporting the KEA algorithm shall, at a minimum, recognize the OID identified in section 3.1.1.

The certificate conveys the KEA public key through the subjectPublicKey field. This subjectPublicKey field is a BIT STRING. Section 3.1.2 specifies the method for encoding a KEA public key as a BIT STRING. Conforming CAs shall encode the KEA public key as described in Section 3.1.2 when issuing certificates containing public keys for the KEA algorithm. Conforming applications supporting the KEA algorithm shall decode the subjectPublicKey as described in section 3.1.2 when the algorithm identifier is the one presented in 3.1.1.

3.1.1. Algorithm Identifier and Parameters

The Key Exchange Algorithm (KEA) is an algorithm for exchanging keys. A KEA "pairwise key" may be generated between two users if their KEA public keys were generated with the same KEA parameters. The KEA parameters are not included in a certificate; instead a "domain identifier" is supplied in the parameters field.

When the subjectPublicKeyInfo field contains a KEA key, the algorithm identifier and parameters shall be as defined in [sdn.701r]:

```
id-keyExchangeAlgorithm  OBJECT IDENTIFIER   ::=
       { 2 16 840 1 101 2 1 1 22 }

KEA-Parms-Id     ::= OCTET STRING
```

CAs shall populate the parameters field of the AlgorithmIdentifier within the subjectPublicKeyInfo field of each certificate containing a KEA public key with an 80-bit parameter identifier (OCTET STRING), also known as the domain identifier. The domain identifier will be computed in three steps: (1) the KEA parameters are DER encoded using the Dss-Parms structure; (2) a 160-bit SHA-1 hash is generated from the parameters; and (3) the 160-bit hash is reduced to 80-bits by performing an "exclusive or" of the 80 high order bits with the 80 low order bits. The resulting value is encoded such that the most significant byte of the 80-bit value is the first octet in the octet string.

The Dss-Parms is provided in [RFC 2459] and reproduced below for completeness.

```
Dss-Parms  ::=  SEQUENCE  {
    p             INTEGER,
    q             INTEGER,
    g             INTEGER  }
```

3.1.2. Encoding of KEA Public Keys

A KEA public key, y, is conveyed in the subjectPublicKey BIT STRING such that the most significant bit (MSB) of y becomes the MSB of the BIT STRING value field and the least significant bit (LSB) of y becomes the LSB of the BIT STRING value field. This results in the following encoding: BIT STRING tag, BIT STRING length, 0 (indicating that there are zero unused bits in the final octet of y), BIT STRING value field including y.

3.2. Key Usage Extension in KEA certificates

The key usage extension may optionally appear in a KEA certificate. If a KEA certificate includes the keyUsage extension, only the following values may be asserted:

keyAgreement;
encipherOnly; and
decipherOnly.

The encipherOnly and decipherOnly values may only be asserted if the keyAgreement value is also asserted. At most one of encipherOnly and decipherOnly shall be asserted in keyUsage extension. Generally, the keyAgreement value is asserted without either the encipherOnly or decipherOnly value being asserted.

4. ASN.1 Modules

4.1 1988 Syntax

```
PKIXkea88 {iso(1) identified-organization(3) dod(6)
         internet(1) security(5) mechanisms(5) pkix(7)
         id-mod(0) id-mod-kea-profile-88(7) }


BEGIN ::=

-- EXPORTS ALL --

-- IMPORTS NONE --
```

```
      id-keyExchangeAlgorithm  OBJECT IDENTIFIER   ::=
             { 2 16 840 1 101 2 1 1 22 }

      KEA-Parms-Id     ::= OCTET STRING

   END
```

4.2 1993 Syntax

```
      PKIXkea93 {iso(1) identified-organization(3) dod(6)
            internet(1) security(5) mechanisms(5) pkix(7)
            id-mod(0) id-mod-kea-profile-93(8) }


      BEGIN ::=

   -- EXPORTS ALL --

   IMPORTS         ALGORITHM-ID
           FROM PKIX1Explicit93 {iso(1) identified-organization(3)
           dod(6) internet(1) security(5) mechanisms(5) pkix(7)
           id-mod(0) id-pkix1-explicit-93(3) }

     KeaPublicKey ALGORITHM-ID ::=  { OID id-keyExchangeAlgorithm
                                      PARMS KEA-Parms-Id }

      id-keyExchangeAlgorithm  OBJECT IDENTIFIER   ::=
             { 2 16 840 1 101 2 1 1 22 }

      KEA-Parms-Id     ::= OCTET STRING

   END
```

5. References

[KEA] "Skipjack and KEA Algorithm Specification", Version 2.0, 29 May 1998. available from http://csrc.nist.gov/encryption/skipjack-kea.htm

[SDN.701R] SDN.701, "Message Security Protocol", Revision 4.0 1996-06-07 with "Corrections to Message Security Protocol, SDN.701, Rev 4.0, 96-06-07." August 30, 1996.

[RFC 2459] Housley, R., Ford, W., Polk, W. and D. Solo "Internet X.509 Public Key Infrastructure: X.509 Certificate and CRL Profile", RFC 2459, January 1999.

6. Security Considerations

This specification is devoted to the format and encoding of KEA keys in X.509 certificates. Since certificates are digitally signed, no additional integrity service is necessary. Certificates need not be kept secret, and unrestricted and anonymous access to certificates and CRLs has no security implications.

However, security factors outside the scope of this specification will affect the assurance provided to certificate users. This section highlights critical issues that should be considered by implementors, administrators, and users.

The procedures performed by CAs and RAs to validate the binding of the subject's identity of their public key greatly affect the assurance that should be placed in the certificate. Relying parties may wish to review the CA's certificate practice statement.

The protection afforded private keys is a critical factor in maintaining security. Failure of users to protect their KEA private keys will permit an attacker to masquerade as them, or decrypt their personal information.

The availability and freshness of revocation information will affect the degree of assurance that should be placed in a certificate.

While certificates expire naturally, events may occur during its natural lifetime which negate the binding between the subject and public key. If revocation information is untimely or unavailable, the assurance associated with the binding is clearly reduced. Similarly, implementations of the Path Validation mechanism described in section 6 that omit revocation checking provide less assurance than those that support it.

The path validation algorithm specified in [RFC 2459] depends on the certain knowledge of the public keys (and other information) about one or more trusted CAs. The decision to trust a CA is an important decision as it ultimately determines the trust afforded a certificate. The authenticated distribution of trusted CA public keys (usually in the form of a "self-signed" certificate) is a security critical out of band process that is beyond the scope of this specification.

In addition, where a key compromise or CA failure occurs for a trusted CA, the user will need to modify the information provided to the path validation routine. Selection of too many trusted CAs will make the trusted CA information difficult to maintain. On the other hand, selection of only one trusted CA may limit users to a closed

community of users until a global PKI emerges.

The quality of implementations that process certificates may also affect the degree of assurance provided. The path validation algorithm described in section 6 relies upon the integrity of the trusted CA information, and especially the integrity of the public keys associated with the trusted CAs. By substituting public keys for which an attacker has the private key, an attacker could trick the user into accepting false certificates.

The binding between a key and certificate subject cannot be stronger than the cryptographic module implementation and algorithms used to generate the signature.

7. Authors' Addresses

Russell Housley
SPYRUS
381 Elden Street
Suite 1120
Herndon, VA 20170
USA

EMail: housley@spyrus.com

Tim Polk
NIST
Building 820, Room 426
Gaithersburg, MD 20899
USA

EMail: wpolk@nist.gov

8. Full Copyright Statement