

IPSEC Working Group
Internet Engineering Task
INTERNET-DRAFT
Expires in six months

John Kennedy
Cylink Corporation
John Marchioni
Cylink Corporation
February 21, 1996

DSS Profile for X.509 Certificates
<[draft-ietf-ipsec-dss-cert-00.txt](#)>

Status of this Memo

This document is a submission to the IETF Internet Protocol Security (IPSEC) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@ans.net) or to the authors.

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Comments should be sent to the IP Security WG mailing list (ipsec@ans.net).

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``[1id-abstracts.txt](#)'' listing contained in the Internet-Drafts Shadow Directories on [ftp.is.co.za](ftp://ftp.is.co.za) (Africa), [nic.nordu.net](ftp://nic.nordu.net) (Europe), [munniari.oz.au](ftp://munniari.oz.au) (Pacific Rim), [ds.internic.net](ftp://ds.internic.net) (US East Coast), or [ftp.isi.edu](ftp://ftp.isi.edu) (US West Coast).

Distribution of this memo is unlimited.

Abstract

This document describes the ASN.1 [\[1\]](#) encoding for an CCITT 1988 [X.509](#) [\[2\]](#) certificate profiled for use with the NIST Digital Signature Standard (DSS) [\[3\]](#).

For details not covered in this document the reader should refer to its base references: X.509 (1993) | ISO/IEC 9594-8 and Amendment 1 to ITU Rec. X.509 (1993) | ISO/IEC 9594-8 : 1995.

[1.](#) ASN.1 Definition of Certificate

The abstract definition of the certificate is as follows:

```
Certificate ::= SIGNED { SEQUENCE {  
    version          [0] Version DEFAULT v1,  
    serialNumber      CertificateSerialNumber,  
    signature         AlgorithmIdentifier,  
    issuer            Name,
```

```

        validity                Validity,
        subject                  Name,
        subjectPublicKeyInfo     SubjectPublicKeyInfo,
        issuerUniqueIdentifier   [1]    IMPLICIT UniqueIdentifier OPTIONAL,
                                     -- if present, must be v2 or v3
        subjectUniqueIdentifier [2]    IMPLICIT UniqueIdentifier OPTIONAL,
                                     -- if present, must be v2 or v3
        extensions               [3]    Extensions OPTIONAL
    }}

Version ::=      INTEGER { 1(0), v2(1), v3(2) }

CertificateSerialNumber ::=      INTEGER

AlgorithmIdentifier ::=      SEQUENCE {
    algorithm      ALGORITHM.&id ({SupportedAlgorithms}),
    parameters     ALGORITHM.&id ({SupportedAlgorithms}{ @algorithm}) OPTI

-- SupportedAlgorithms          ALGORITHM          ::=      {...|...}

-- DSA Signature Algorithm

-- The Digital Signature Algorithm (DSA) is also called the Digital Signature
-- Standard (DSS). DSA
-- was developed by the U.S. Government, and DSA is used in conjunction with th
-- way hash function (SHA-1 is described in FIPS 180-1). DSA is described in F
-- The ASN.1 object identifier used to identify this signature algorithm is:

        dsaWithSHA-1 OBJECT IDENTIFIER ::= {
            joint-iso-ccitt(2) country(16) US(840) organization(1) us-govern
            infosec(1) algorithms(1) dsa-sha1 (2) }

-- DSA Parameters

- When this object identifier is used with the ASN.1 type AlgorithmIdentifier,
- component of that type is optional. If it is absent, the DSA parameters p, q
- be known, otherwise the parameters are included using the following ASN.1 str

        Dss-Parms ::= SEQUENCE {
            p      OCTET STRING,
            q      OCTET STRING,
            g      OCTET STRING }

-- DSA Signature Block

-- Prior to the bitstring encoding of the certificate issuers DSA signature the
-- be encoded using the distinguished rules as follows:

Dss-sig ::= SEQUENCE {
    r      OCTET STRING,
    s      OCTET STRING }

Validity ::=      SEQUENCE {

```

```

        notBefore      UTCTime,
        notAfter       UTCTime }

SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm      AlgorithmIdentifier,
    subjectPublicKey BIT STRING }

```

2. Certificate Extensions

The standard extensions are described in Amendment 1 to ITU Rec. X.509 (1993) | ISO/IEC 9594-8 : 1995. A subset of extension will need to be chosen for this profile. The extensions field allows addition of new fields to the certificate structure without modification to the ASN.1 definition. An extension field consists of an extension object identifier, a criticality flag, and a canonical encoding of a data value of an ASN.1 type associated with the object identifier already specified.

When a system processes a certificate but does not recognize an extension, if the criticality flag is FALSE, the extension may be ignored and the remainder of the certificate information may be processed as valid. If the criticality flag is TRUE, an unrecognized extension shall cause the system to consider the entire certificate invalid.

3. An overview of the use of the Distinguished Encoding Rules (DER) in Certificate Signature Operations.

(1) Sign; The signing application converts the abstract value (or internal representation) of the certificate information into a bit representation using the DER and signs that bit representation. The signature is then appended onto the abstract value, and both values are then BER (Basic Encoding Rules) encoded to provide a transfer syntax. The same encoder used to apply the DER may be used to apply the transfer syntax, so the transfer syntax can also follow the DER.

(2) Authenticate; The authenticating application will decode the received certificate (containing the certificate information and issuer signature). This application will then have an abstract value for both the certificate information and a signature. The application will then take the resulting abstract value of the certificate information and re-encode it using the DER to produce the same bit representation that was signed. The received signature can now be authenticated using the exact bitstring representation used in the signing operation.

When the DER are applied to information, before that information is signed, the authentication operation (also applying the DER) will always detect if that information has been modified and the incidence of false authentication failures is greatly reduced.

4. Security Considerations

Security issues are not discussed in this document

5. References

[1] CCITT Recommendation X.208 (1992), "Abstract Syntax Notation One"

[2] CCITT Recommendation X.509 (1988), "The Directory - Authentication Framework"

[3] FIPS 186 Digital Signature Standard

Author's Address(es)

Questions about this can be directed to:

John Kennedy
CYLINK Corporation
jkennedy@cylink.com
408-735-5885

John Marchioni
CYLINK Corporation
johnmarc@cylink.com
408-735-5800