Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Text accompanying comment on certificate format**
Date Submitted	**2001-03-06**
Source(s)	Carl Eklund Nokia Voice: +358407499036 Fax: +358943766851 mailto:carl.eklund@nokia.com
Re:	IEEE P802.16/D2
Abstract	Text to be included in the standards document
Purpose	Specify the content of the X.509 Certificates.
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate text contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Sections on certificate format

Carl Eklund
Nokia

0.1 Certificat profile

0.1.1 Certificate format

This section describes the X.509 version 3 certificate format and certificate extensions used in IEEE 802.16 compliant SSs. Table 1 below summarizes the basic fields of an X.509 Version 3 certificate.

Table 1~~X.~~509 Basic Certificate Fields

X.509 v3 Field	Description
tbsCertificate.version	Indicates the X.509 certificate version. Always set to v3 (value of 2)
tbsCertificate.serial-Number	Unique integer the issuing CA assigns to the certificate.
tbsCertificate.signa-ture	OID and optional parameters defining algorithm used to sign the cer-tificate. This field MUST contain the same algorithm identifier as the signatureAlgorithm field below.
tbsCertificate.issuer	Distinguished Name of the CA that issued the certificate
tbsCertificate.validity	Specifies when the certificate becomes active and when it expires.
tbsCertificate.subject	Distinguished Name identifying the entity whose public key is certi-fied in the sub-jectpublic key information field.
tbsCertificate.subject-PublicKeyInfo	Field contains the public key material (public key and parameters) and the identi-fier of the algorithm with which the key is used.
tbsCertificate.issuerU-niqueID	Optional field to allow reuse of issuer names over time.
tbsCertificate.subjec-tUnique ID	Optional field to allow reuse of subject names over time.
tbsCertificate.exten-sions	The extension data.
signatureAlgorithm	OID and optional parameters defining algorithm used to sign the cer-tificate. This field MUST contain the same algorithm identifier as the signature field in tbsCer-tificate.
signatureValue	Digital signature computed upon the ASN.1 DER encoded tbsCertifi-cate.

All certificates and CRLs described in this specification MUST be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [RSA1]; SHA-1 is described in [FIPS-180-1]. Restrictions posed on the certificate values are described below:

0.1.1.1 tbsCertificate.validity.notBefore and tbsCertificate.validity.notAfter

SS certificates will not be renewable, and, thus, must have a validity period greater than the operational lifetime of the SS. A Manufacturer CA certificate's validity period SHOULD exceed that of the SS certificates it issues. The IEEE 802.16 Root CA certificate shall be valid from <date to be determined> for a period to be determined , and exceeding the validity period of the Manufacturer CA certificates it issues. The validity period of a SS certificate MUST begin with the device's data of manufacture; the validity period SHOULD extend out to at least <number of years TBD> years after that manufacturing date. Validity periods MUST be encoded as UTCTime. UTCTime values MUST be expressed Greenwich Mean Time (Zulu) and MUST include seconds (i.e., times are YYMMDDHHMMSSZ), even where the number of seconds is zero.

0.1.1.2 tbsCertificate.serialNumber

Serial numbers for SS certificates signed by a particular issuer MUST be assigned by the manufacturer in increasing order. Thus, if the tbsCertificate.validity.notBefore field of one certificate is greater than the tbsCertificate.validity.notBefore field of another certificate, then the serial number of the first certificate must be greater than the serial number of the second certificate. The Manufacturer SHOULD NOT impose or assume a relationship between the serial number of the certificate and the serial number of the modem to which the certificate is issued.

0.1.1.3 tbsCertificate.signature and signatureAlgorithm

All certificates and CRLs described in this specification MUST be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [RSA1]; SHA-1 is described in [FIPS-180-1]. The ASN.1 OID used to identify the "SHA-1 with RSA" signature algorithm is:

```
sha-1WithRSAEncryption OBJECT IDENTIFIER ::=
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}
```

When the sha-1WithRSAEncryption OID appears within the ASN.1 type AlgorithmIdentifier, as is the case with both tbsCertificate.signature and signatureAlgorithm, the parameters component of that type is the ASN.1 type NULL.

0.1.1.4 tbsCertificate.issuer and tbsCertificate.subject

X.509 Names are SEQUENCES of RelativeDistinguishedNames, which are in turn SETs of AttributeTypeAndValue. AttributeTypeAndValue is a SEQUENCE of an AttributeType (an OBJECT IDENTIFIER) and an AttributeValue. The value of the countryName attribute MUST be a 2-character PrintableString, chosen from ISO 3166; all other AttributeValues MUST be encoded as either T.61/TeletexString or PrintableString character strings. The PrintableString encoding MUST be used if the character string contains only characters from the PrintableString set. Specifically:

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
'()+,-./:=? and space.
```

The T.61/TeletexString MUST be used if the character string contains other characters. The following OIDs are needed for defining issuer and subject Names in PKM certificates:

```
id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}
```

id-at-commonName OBJECT IDENTIFIER ::= {id-at 3}
id-at-countryName OBJECT IDENTIFIER ::= {id-at 6}
id-at-localityName OBJECT IDENTIFIER ::= {id-at 7}
id-at-stateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8}
id-at-organizationName OBJECT IDENTIFIER ::= {id-at 10}
id-at-organizationalUnitName OBJECT IDENTIFIER ::= {id-at 11}

The following subsections describe the attributes which comprise the subject Name forms for each type of PKM certificate. Note that the issuer name form is the same as the subject of the issuing certificate. Additional attribute values that are present, but not specified in the following forms SHOULD NOT cause a device to reject the certificate.2

0.1.1.4.1 Root Certificate

countryName=US
organizationName=TBD
organizationalUnitName=TBD
commonName=TBD

The countryName, organizationName, organizationalUnitName and commonName attributes MUST be included and MUST have the values shown. Other attributes are not allowed and MUST NOT be included.

0.1.1.4.2 Manufacturer Certificate

countryName=<Country of Manufacturer>
[stateOrProvinceName=<state/privince>]
[localityName=<City>]
organizationName=<Company Name>
organizationalUnitName=XX
[organizationalUnitName=<Manufacturing Location>]
commonName=<Company Name> Root Certificate Authority

The countryName, organizationName, and commonName attributes MUST be included and MUST have the values shown. The organizationalUnitName having the value “XX” MUST be included The organizationalUnitName representing manufacturing location SHOULD be included. If included, it MUST be preceded by the organizationalUnitName having value “XX.” The stateOrProvinceName and localityName MAY be included .Other attributes are not allowed and MUST NOT be included.

0.1.1.4.3 SS Certificate

countryName=<Country of Manufacturer>
organizationName=<Company Name>
organizationalUnitName=<manufacturing location>
commonName=<Serial Number>
commonName=<MAC Address>

To distinguish between the two commonNames, the commonName representing the “Serial Number” MUST precede the commonName representing “MAC Address”. Use of the Serial Number field is deprecated. If used, the Serial Number MUST be a unique SS identifier, but MAY be different from the serial

number encoded in thePKM attributes. The MAC address in the SS Certificate MUST be the same as the MAC address in the PKM Attributes. The characters employed in the PrintableString representation of SS serial numbers MUST be restricted to the following character subset:

A-Z (0x41-0x5A)
a-z (0x61-0x7A)
0-9 (0x30-0x39)
"–" (0x2D)

The MAC Address is expressed as six pairs of hexadecimal digits separated by colons (:), e.g., "00:60:21:A5:0A:23". The Alpha HEX characters (A-F) MUST be expressed as uppercase letters.

The organizationalUnitName in a SS certificate, which describes the modem's manufacturing location, SHOULD be the same as the organizationalUnitName in the issuer Name describing a manufacturing location. The countryName, organizationName, organizationalUnitName, and two commonName attributes MUST be included. Other attributes are not allowed and MUST NOT be included.

0.1.1.5 tbsCertificate.subjectPublicKeyInfo

The tbsCertificate.subjectPublicKeyInfo field contains the public key and the public key algorithm identifier. The RSA public key in the SS Certificate MUST be the same as the RSA public key in the PKM Attributes. The tbsCertificate.subjectPublicKeyInfo.algorithm field is an AlgorithmIdentifier structure. The AlgorithemIdentifier's algorithm MUST be RSA encryption, identified by the following OID:

pkcs-1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) 1}
rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1}

The AlgorithmIdentifier's parameters field MUST have ASN.1 type NULL. The RSA public key shall be encoded using the ASN.1 type RSAPublicKey:

RSAPublicKey ::= SEQUENCE {
modulus INTEGER, -- n
publicExponent INTEGER, -- e -- }

where modulus is the modulus n, and publicExponent is the public exponent e. The DER encoded RSAPublicKey is the value of the BIT STRING tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.

0.1.1.6 tbsCertificate.issuerUniqueID and tbsCertificate.subjectUniqueID

The issuerUniqueID and subjectUniqueID fields MUST be omitted for all three of PKM's certificate types.

0.1.1.7 tbsCertificate.extensions

0.1.1.7.1 SS Certificates

SS certificates MAY contain noncritical extensions; they MUST NOT contain critical extensions. If the KeyUsage extension is present, the keyAgreement and keyEncipherment bits MUST be turned on, keyCertSign and cRLSign bits MUST be turned off, and all other bits SHOULD be turned off.

0.1.1.7.2 Root and Manufacturer Certificates

Root and Manufacturer certificates MAY contain the Basic Constraints extension. If included, the Basic Constraints extension may appear as a critical extension or as a noncritical extension. Root and Manufacturer certificates MAY contain noncritical extensions; they MUST NOT contain critical extensions other than, possibly, the Basic Constraints extension. If the KeyUsage extension is present in a Root or Manufacturer certificate the keyCertSign bit MUST be turned on and all other bits SHOULD be turned off.

0.1.1.8 signatureValue

In all three PKM certificate types, the signatureValue contains the RSA (with SHA-1) signature computed over the ASN.1 DER encoded tbsCertificate. The ASN.1 DER encoded tbsCertificate is used as input to the RSA signature function. The resulting signature value is ASN.1 encoded as a BIT STRING and included in the Certificate's signatureValue field.

0.1.2 SS Certificate Storage and Management in the SS

Manufacturer-issued SS certificates MUST be stored in SS permanent, write-once memory. SSs that have factory-installed RSA private/public key pairs MUST also have factory-installed SS certificates. SSs that rely on internal algorithms to generate an RSA key pair MUST support a mechanism for installing a manufacturer-issued SS certificate following key generation.The Root CA's (RSA) public key MUST be placed into SS's non-volatile memory. The CA certificate of the Manufacturer CA that signed the SS certificate MUST be embedded into the SS software. If a manufacturer issues SS certificates with multiple Manufacturer CA certificates, the SS software must include ALL of that manufacturer's CA certificates. The specific Manufacturer CA certificate installed by the SS (i.e., advertised in Authentication Information messages and returned by the MIB object) will be that identifying the issuer of that modem's SS certificate.

0.1.3 Certificate Processing and Management in the BS

PKM employs digital certificates to allow BSs to verify the binding between a SS's identity (encoded in an X.509 digital certificate's subject names) and its public key. The BS does this by validating the SS certificate's certification path or chain. This path wil typically consist of three chained certificates: starting with the SS Certificate, the path leads to the certificate of the Manufacturer CA that issued the SS Certificate, and ends at the Root CA's self-signed certificate. Validating the chain means verifying the Manufacturer CA Certificate's signature with the Root CA's public key and then verifying the SS Certificate's signature with the public key of the Manufacturer CA.