Network Working Group
Internet Draft
Intended Status: Standard Track
Updates: RFC 5208 (once approved)
Expires: 20 April 2010

Sean Turner, IECA
20 October 2009

Asymmetric Key Packages
draft-turner-asymmetrickeyformat-02.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the syntax for private key information and a content type for it. Private-key information includes a private key for a specified public-key algorithm and a set of attributes. The Cryptographic Message Syntax (CMS), as defined in RFC 3852, can be used to digitally sign, digest, authenticate, or encrypt the asymmetric key format content type. This document updates RFC 5208.

1. Introduction

This document defines the syntax for private key information and a Cryptographic Message Syntax (CMS) [RFC5652] content type for it. Private-key information includes a private key for a specified public-key algorithm and a set of attributes. The CMS can be used to digitally sign, digest, authenticate, or encrypt the asymmetric key format content type. This document updates PKCS#8 v1.2 [RFC5208] sections 5 and 7, and it adds two new sections; the first covers protecting the Asymmetric Key Content Type, and the second discusses compatibility with other private-key formats.

1.1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2. ASN.1 Syntax Notation

The key package is defined using ASN.1 [X.680], [X.681], [X.682], and [X.683].

1.3. Summary of Updates to RFC 5208

The following summarizes the updates to [RFC5208]:

- Changed the name "PrivateKeyInfo" to "OneAsymmetricKey". This reflects the addition of the public key field to allow both parts of the asymmetric key to be conveyed separately. Not all algorithms will use both fields; however, the publicKey field was added for completeness.

- Defined Asymmetric Key Package CMS content type.

- Removed redundant IMPLICIT from attributes.

- Added publicKey to OneAsymmetricKey and updated the version number.

- Added that PKCS#9 attributes may be supported.

- Added discussion of compatibility with other private-key formats.

- Added requirements for encoding rule set.

- Changed imports from PKCS#5 to [RFCTBD3].

2. Asymmetric Key Package CMS Content Type

This section updates section 5 of [RFC5208].

The asymmetric key package CMS content type is used to transfer one or more plaintext asymmetric keys from one party to another. An asymmetric key package MAY be encapsulated in one or more CMS protecting content types (see Section 4). Earlier versions of this specification [RFC5208] did not specify a particular encoding rule set, but generators SHOULD use DER [X.690] and receivers SHOULD be prepared to handle BER [X.690] and DER [X.690].

The asymmetric key package content type has the following syntax:

```
PKCS7-CONTENT-TYPE ::= TYPE-IDENTIFIER

asymmetric-key-package PKCS7-CONTENT-TYPE ::=
  { AsymmetricKeyPackage IDENTIFIED BY id-ct-KP-aKeyPackage }

id-ct-KP-aKeyPackage OBJECT IDENTIFIER ::=
  { TBD }

AsymmetricKeyPackage ::= SEQUENCE SIZE (1..MAX) OF OneAsymmetricKey

OneAsymmetricKey ::= SEQUENCE {
  version                   Version,
  privateKeyAlgorithm       PrivateKeyAlgorithmIdentifier,
  privateKey                PrivateKey,
  attributes            [0] Attributes OPTIONAL,
  publicKey             [1] PublicKey OPTIONAL
}

PrivateKeyInfo ::= OneAsymmetricKey

-- PrivateKeyInfo is used by [P12].  If version is set to 1,
-- publicKey MUST be absent. When v1, PrivateKeyInfo is the same
-- as it was in [RFC5208].
```

```
     Version ::= INTEGER { v1(0), v2(1) } (v1, v2,...)

     PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
                                    { { PrivateKeyAlgorithms } }

     PrivateKey ::= OCTET STRING
                        -- Content varies based on type of key. The
                        -- algorithm identifier dictates the format of
                        -- the key.

     PublicKey ::= BIT STRING
                       -- Content varies based on type of key. The
                       -- algorithm identifier dictates the format of
                       -- the key.

     Attributes ::= Set of Attribute
```

The AsymmetricKeyPackage contains one or more OneAsymmetricKey elements.

The syntax of OneAsymmetricKey accommodates a version number, an indication of the asymmetric algorithm to be used with the private key, a private key, optional keying material attributes (e.g., userCertificate from [X.520]), and an optional public key. In general, either the public key or the certificate will be present. In very rare cases will both the public key and the certificate be present as this includes two copies of the public key. OneAsymmetricKey is a renamed extension of the PrivateKeyInfo syntax defined in [RFC5208]. The new name better reflects the ability to carry both private and public key components. Backwards compatibility with the original PrivateKeyInfo is preserved via version number. The fields in OneAsymmetricKey are used as follows:

- version identifies the version of OneAsymmetricKey. If publicKey is present, then version is set 2 else version is set to 1.

- privateKeyAlgorithm identifies the private-key algorithm and optionally contains parameters associated with the asymmetric key. The algorithm is identified by an object identifier (OID) and the format of the parameters depends on the OID. The value placed in privateKeyAlgorithmIdentifier is the value an originator would apply to indicate which algorithm is to be used with the private key.

- privateKey is an OCTET STRING whose contents are the value of the private key. The interpretation of the contents is defined in the registration of the private-key algorithm. For example, a DSA key is an INTEGER, an RSA key is represented as RSAPrivateKey as

defined in [RFC3447], and an ECC key is represented as ECPrivateKey as defined in [RFCTBD2].

- attributes is optional. It contains information corresponding to the public key (e.g., certificates). The attributes field uses the class ATTRIBUTE which is restricted by the SupportedAttributes parameterized type. SupportedAttributes is an open ended set in this document. Others documents can constrain these values. Attributes from [RFC2985] MAY be supported.

- publicKey is optional. When present, it contains the public key encoded as an OCTET STRING. The structure within the octet string, if any, depends on the privateKeyAlgorithm. For example, a DSA key is an INTEGER. Other documents may define additional private key formats. Note that RSA public keys are included in RSAPrivateKey (i.e., n and e are present), as per [RFC3447], and ECC public keys are included in ECPrivateKey (i.e., in the publicKey field), as per [RFCTBD2].

3. Protecting the AsymmetricKeyPackage

CMS protecting content types, [RFC5652] and [RFC5083], can be used to provide security to the AsymmetricKeyPackage:

- SignedData can be used to apply a digital signature to the AsymmetricKeyPackage.

- EncryptedData can be used to encrypt the AsymmetricKeyPackage to provide confidentiality but does not distribute the content encryption keys.

- EnvelopedData can be used to encrypt the AsymmetricKeyPackage with simple symmetric encryption, where the sender and the receiver already share the necessary encryption key.

- AuthenticatedData can be used to protect the AsymmetricKeyPackage with message authentication codes, where key management information is handled in a manner similar to EnvelopedData.

- AuthEnvelopedData can be used to protect the AsymmetricKeyPackage with algorithms that support authenticated encryption, where key management information is handled in a manner similar to EnvelopedData.

4. Other Private-Key Format Considerations

This document defines the syntax and the semantics for a content type that exchanges asymmetric private keys. There are two other formats that have been used for the transport of asymmetric private keys:

- Personal Information Exchange (PFX) Syntax Standard [P12], which is more commonly referred to as PKCS #12 or simply P12, is a transfer syntax for personal identity information, including private keys, certificates, miscellaneous secrets, and extensions. OneAsymmetricKey, PrivateKeyInfo, and EncryptedPrivateKeyInfo [RFC5208] can be carried in a P12 message. The private key information, OneAsymmetricKey and PrivateKeyInfo, are carried in the P12 keyBag BAG-TYPE. EncryptedPrivateKeyInfo is carried in the P12 pkcs8ShroudedKeyBag BAG-TYPE. In current implementations, the file extensions .pfx and .p12 can be used interchangeably.

- Microsoft's private key proprietary transfer syntax. The .pvk file extension is used for local storage.

The .pvk and .p12/.pfx formats are not interchangeable; however, conversion tools exist to convert from one format to another.

[RFCTBD3] defines the appication/pkcs8 media type and .p8 file extension.

To extract the private key information from the AsymmetricKeyPackage, the encapsulating layers need to be removed. At a minimum, the outer ContentInfo [RFC5652] layer needs to be removed. If the AsymmetricKeyPackage is encapsulated in a SignedData [RFC5652], then the SignedData and EncapsulatedContentInfo layers [RFC5652] also need to be removed. The same is true for EnvelopedData, EncryptedData, and AuthenticatedData all from [RFC5652] as well as AuthEnvelopedData from [RFC5083]. Once all the outer layers are removed, there are as many sets of private key information as there are OneAsymmetricKey structures. OneAsymmetricKey and PrivateKeyInfo are the same structure; therefore, either can be saved as a .p8 file or copied in to the P12 KeyBack BAG-TYPE. Removing encapsulating security layers will invalidate any signature and may expose the key to unauthorized disclosure.

.p8 files are sometimes PEM encoded. When .p8 files are PEM encoded they use the .pem file extension. PEM encoding is the Base64 encoding [RFC2045] of either the DER encoded EncryptedPrivateKeyInfo sandwiched between:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
-----END ENCRYPTED PRIVATE KEY-----
```

or the PrivateKeyInfo or the OneAsymmetricKey sandwiched between:

```
-----BEGIN PRIVATE KEY-----
-----END PRIVATE KEY-----
```

5. Security Considerations

The security considerations in [RFC5208] also apply to this document.

The asymmetric key package contents are not protected. This content type can be combined with a security protocol to protect the contents of the package.

6. IANA Considerations

None: All identifiers are already registered. Please remove this section prior to publication as an RFC.

7. References

7.1. Normative References

[RFC2045] Freed, .N, and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 5652, September 2009.

[RFC5208] Kaliski, B., "PKCS #8: Private Key Information Syntax Standard Version 1.2", RFC 5208, May 2008.

[RFCTBD1] Schaad, J., and P. Hoffman, "New ASN.1 Modules for PKIX", draft-ietf-pkix-new-asn1-07.txt, work-in-progress.

[RFCTBD3] Jennings C., and J. Fischl "Certificate Management Service for The Session Initiation Protocol (SIP)", draft-ietf-sip-certs-09.txt, work-in-progress.

[X.680] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002. Information Technology - Abstract Syntax Notation One.

[X.681] ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002. Information Technology - Abstract Syntax Notation One: Information Object Specification.

[X.682] ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002. Information Technology - Abstract Syntax Notation One: Constraint Specification.

[X.683] ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002. Information Technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications.

[X.690] ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002. Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

7.2. Non-Normative References

[P12] RSA Laboratories, "PKCS #12 v1.0: Personal Information Exchange Syntax", June 1999.

[RFC2985] Nystrom, M., and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, November 2000.

[RFC3447] Jonsson, J., and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC5083] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, November 2007.

[X.520] ITU-T Recommendation X.520 (2005) | ISO/IEC 9594-6:2005, Information technology - Open Systems Interconnection - The Directory: Selected attribute types.

[RFCTBD2] Turner, S., and D. Brown, "EC Private Key Info Structure", draft-turner-ecprivatekey-00.txt, work-in-progress.

APPENDIX A: ASN.1 Module

This annex provides the normative ASN.1 definitions for the structures described in this specification using ASN.1 as defined in [X.680] through [X.683].

```
AsymmetricKeyPackageModulev1  { tbd }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL
```

```
   IMPORTS NOTHING

   Attribute{}, ATTRIBUTE
     FROM PKIX-CommonTypes-2009  -- FROM [RFCTBD1]
       { iso(1) identified-organization(3) dod(6) internet(1)
         security(5) mechanisms(5) pkix(7) id-mod(0)
         id-mod-pkixCommon-02(57) }

   AlgorithmIdentifier{}
     FROM AlgorithmInformation-2009  -- FROM [RFCTBD1]
       { iso(1) identified-organization(3) dod(6) internet(1)
         security(5) mechanisms(5) pkix(7) id-mod(0)
         id-mod-algorithmInformation-02(58) }

   ;

   PKCS7-CONTENT-TYPE ::= TYPE-IDENTIFIER

   KeyPackageContentTypes PKCS7-CONTENT-TYPE ::= {
     asymmetric-key-package,
     ... -- Expect additional content types --
   }

   asymmetric-key-package PKCS7-CONTENT-TYPE ::=
     { AsymmetricKeyPackage IDENTIFIED BY id-ct-KP-aKeyPackage }

   id-ct-KP-aKeyPackage OBJECT IDENTIFIER ::=
     { TBD }

   AsymmetricKeyPackage ::= SEQUENCE SIZE (1..MAX) OF OneAsymmetricKey

   OneAsymmetricKey ::= SEQUENCE {
     version                   Version,
     privateKeyAlgorithm       PrivateKeyAlgorithmIdentifier,
     privateKey                PrivateKey,
     attributes            [0] Attributes OPTIONAL,
     publicKey             [1] PublicKey OPTIONAL
   }

   PrivateKeyInfo ::= OneAsymmetricKey

   -- PrivateKeyInfo is used by [P12].  If version is set to 1,
   -- publicKey MUST be absent. When v1, PrivateKeyInfo is the same
   -- as it was in [RFC5208].

   Version ::= INTEGER {v1(0), v2(1)} (v1, v2,...)

   PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier
                                      { { PrivateKeyAlgorithms } }
```

```
   PrivateKey ::= OCTET STRING
                    -- Content varies based on type of key. The
                    -- algorithm identifier dictates the format of
                    -- the key.

   PublicKey ::= BIT STRING
                    -- Content varies based on type of key. The
                    -- algorithm identifier dictates the format of
                    -- the key.

   Attributes ::= Set of Attribute { { SupportedAttributes } }

   SupportedAttributes ATTRIBUTE :: {
     ... -- For local profiles
   }

   EncryptedPrivateKeyInfo ::= SEQUENCE {
     encryptionAlgorithm  EncryptionAlgorithmIdentifier,
     encryptedData        EncryptedData }

   EncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
                                    { { KeyEncryptionAlgorithms } }

   EncryptedData ::= OCTET STRING -- Encrypted PrivateKeyInfo

   PrivateKeyAlgorithms ALGORITHM-IDENTIFIER ::= {
     ... -- Extensible
   }

   KeyEncryptionAlgorithms ALGORITHM-IDENTIFIER ::= {
     ... -- Extensible
   }

   END
```

Acknowledgements

Many thanks go out to the Burt Kaliski and Jim Randall at RSA. Without the prior version of the document, this one wouldn't exist.

We'd also like to thank Pasi Eronen, Russ Housley, and Carl Wallace.

Author's Address

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

Email: turners@ieca.com