

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 18 February 2026

P. Gutmann
University of Auckland
17 August 2025

PKCS #15 Updates
draft-gutmann-pkcs15-03

Abstract

This document describes updates to the PKCS #15 standard made since the original publication of the standard.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 18 February 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this

material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	2
2. ValidFrom/ValidTo Dates	2
3. Key Identifiers	3
4. Authenticated-Enveloped-Data	4
5. Public/Private Key Binding	5
6. IANA Considerations	6
7. Security Considerations	7
8. Normative References	7
Author's Address	7

1. Introduction

After the publication of the original PKCS #15 standard [PKCS15v1] it saw minor updates that were only published as drafts and circulated informally among implementers [PKCS15v1_1][PKCS15v1_2], but never finalised due to the dissolution of the organisation that published the standards. Since the standard, including the updates, remains in active use today, this document gathers the updates in a single location for reference by implementers.

The updates cover the extension of the original PKCS #15 identifiers to handle validFrom/validTo dates for certificates and PGP/OpenPGP keys and email addresses, the addition of support for CMS Authenticated-Enveloped-Data, and support for cryptographic binding of public-key components to private-key ones.

Since names and definitions have changed across the different drafts (for example v1.0 used PKCS15XXXXAttributes while v1.1 and later used XXXXAttributes), this document uses the v1.1 ASN.1 module pkcs-15v1_1.asn as its baseline.

2. ValidFrom/ValidTo Dates

PKCS #15 v1.0 didn't support the validFrom/validTo dates that are required for certificates, these were added in the PKCS #15 v1.2 draft and extend the CommonCertificateAttributes to add:

```

    validFrom          GeneralizedTime OPTIONAL,
    validTo            [4] GeneralizedTime OPTIONAL,

```

For context, the full CommonCertificateAttributes are then:

```

CommonCertificateAttributes ::= SEQUENCE {
    id                Identifier,
    authority          BOOLEAN DEFAULT FALSE,
    identifier         CredentialIdentifier {{KeyIdentifiers}} OPTIONAL,
    certHash          [0] OOBCertHash OPTIONAL,
    ...,
    trustedUsage [1] Usage OPTIONAL,
    identifiers       [2] SEQUENCE OF CredentialIdentifier{{KeyIdentifiers}} OP
TIONAL,
    implicitTrust [3] BOOLEAN DEFAULT FALSE,
    validFrom       GeneralizedTime OPTIONAL,
    validTo         [4] GeneralizedTime OPTIONAL,
}

```

3. Key Identifiers

PKCS #15 v1.0, designed for use with smart cards, didn't support PGP/OpenPGP or email use, making it difficult to implement PKCS #11 with PKCS #15 as the storage format. The PKCS #15 v1.2 draft extended CredentialIdentifier to include these additional IDs, which extend the existing KeyIdentifiers values to add pgp, openPGP, and uri identifiers:

```

KeyIdentifiers KEY-IDENTIFIER ::= {
    issuerAndSerialNumber |
    issuerAndSerialNumberHash |
    subjectKeyId |
    subjectKeyHash |
    issuerKeyHash |
    issuerNameHash |
    subjectNameHash |
    pgp |
    openPGP |
    uri
    ...
}

pgp KEY-IDENTIFIER ::=
    {SYNTAX OCTET STRING SIZE(8) IDENTIFIED BY 8}
    -- RFC 4880 V3 (PGP 2.x) key ID

openPGP KEY-IDENTIFIER ::=
    {SYNTAX OCTET STRING SIZE(8) IDENTIFIED BY 9}
    -- RFC 4880 V4 key ID

uri KEY-IDENTIFIER ::=
    {SYNTAX UTF8String IDENTIFIED BY 10}
    -- Typically email address but may be a more general URI

```

4. Authenticated-Enveloped-Data

PKCS #15 v1.0 predates the existence of CMS Authenticated-Enveloped-Data, which was added in the PKCS #15 v1.2 draft by extending the ObjectValue/ PathOrObjects CHOICE to include a new content type AuthEnvelopedData alongside the existing EnvelopedData. For ObjectValue this is:

```
direct-protected-auth [4] AuthEnvelopedData {Type},
```

For PathOrObjects this is:

```
direct-protected-auth [4] AuthEnvelopedData {SEQUENCE OF ObjectType},
```

Note that the tags jump from the v1.1 'direct-protected [2] EnvelopedData' to 'direct-protected-auth [4] AuthEnvelopedData', the [3] tag was used for another object type whose purpose is now lost. For context, the full ObjectValue / PathOrObjects are then:

```

ObjectValue { Type } ::= CHOICE {
    indirect                ReferencedValue {Type},
    direct                  [0] Type,
    indirect-protected      [1] ReferencedValue {EnvelopedData {Type}},
    direct-protected        [2] EnvelopedData {Type},
    direct-protected-auth   [4] AuthEnvelopedData {Type}
}

PathOrObjects {ObjectType} ::= CHOICE {
    path                    Path,
    objects                  [0] SEQUENCE OF ObjectType,
    ...,
    indirect-protected      [1] ReferencedValue {EnvelopedData {SEQUENCE OF O
bjecType}},
    direct-protected        [2] EnvelopedData {SEQUENCE OF ObjectType}
    direct-protected-auth   [4] AuthEnvelopedData {SEQUENCE OF ObjectType},
}

```

5. Public/Private Key Binding

An update to the PKCS #15 v1.2 draft provided for cryptographic binding between the private key and public key data. This protects the otherwise typically unprotected public-key objects from undetectable manipulation. This cryptographic binding is added by extending the existing privateXXXKey types with new privateXXXKeyExt types that include the cryptographic binding:

```

PrivateKeyType ::= CHOICE {
    privateRSAKey PrivateKeyObject {PrivateRSAKeyAttributes},
    privateECKKey  [0] PrivateKeyObject {PrivateECKKeyAttributes},
    privateDHKey   [1] PrivateKeyObject {PrivateDHKeyAttributes},
    privateDSAKKey [2] PrivateKeyObject {PrivateDSAKKeyAttributes},
    privateKEAKKey [3] PrivateKeyObject {PrivateKEAKKeyAttributes},
    privateRSAKeyExt [4] PrivateKeyObject {PrivateRSAKeyAttributesExt},
    privateECKKeyExt [5] PrivateKeyObject {PrivateECKKeyAttributesExt},
    privateDSAKKeyExt [6] PrivateKeyObject {PrivateDSAKKeyAttributesExt},
    ...
}

```

The Ext variants wrap the original XXXPrivateKeyObject in an additional SEQUENCE that adds an [ESSCertIDv2] field, with the ESSCertIDv2 restricted to contain only a SHA-2 hash of the public key data in SubjectPublicKeyInfo form. In other words the ESSCertIDv2:

```

ESSCertIDv2 ::= SEQUENCE {
    hashAlgorithm AlgorithmIdentifier DEFAULT {algorithm id-sha256},
    certHash      Hash,
    issuerSerial   IssuerSerial OPTIONAL
}

```

is present as:

```
ESSCertIDv2 ::= SEQUENCE {
    certHash          OCTET STRING SIZE(32),
}
```

The resulting XXXPrivateKeyObject is then, in an ASN.1-like notation:

```
XXXPrivateKeyObject ::= SEQUENCE {
    spkiHash          ESSCertIDv2,
    -- Followed by the original PrivateXXXKeyObject
}
```

For example for an ECC private key the original:

```
ECPrivateKey ::= INTEGER
```

would become in extended form with cryptographic binding:

```
ECPrivateKeyExt ::= SEQUENCE {
    spkiHash          ESSCertIDv2,
    value             INTEGER
}
```

For an RSA private key the original:

```
RSAPrivateKeyObject ::= SEQUENCE {
    modulus           [0] INTEGER OPTIONAL, -- n
    ...
    coefficient       [7] INTEGER OPTIONAL -- inv(q) mod p
}
```

would become in extended form with cryptographic binding:

```
RSAPrivateKeyObjectExt ::= SEQUENCE {
    spkiHash          ESSCertIDv2,
    value             SEQUENCE {
        modulus       [0] INTEGER OPTIONAL, -- n
        ...
        coefficient   [7] INTEGER OPTIONAL -- inv(q) mod p
    }
}
```

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

This document serves to document minor updates to the original PKCS #15 standard, there are no security considerations present beyond those in the original standard.

8. Normative References

[ESSCertIDv2]

Schaad, J., "Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility", RFC 5035, August 2007, <<http://www.ietf.org/rfc/rfc5035.txt>>.

[PKCS15v1] Laboratories, R., "PKCS #15 v1.0: Cryptographic Token Information Format Standard", PKCS 15, April 1999.

[PKCS15v1_1]

Laboratories, R., "PKCS #15 v1.1: Cryptographic Token Information Syntax Standard (draft)", PKCS 15, December 1999.

[PKCS15v1_2]

Laboratories, R., "PKCS #15 v1.2: Cryptographic Token Information Syntax Standard (draft)", PKCS 15, March 2010.

Author's Address

Peter Gutmann
University of Auckland
Department of Computer Science
Auckland
New Zealand
Email: pgut001@cs.auckland.ac.nz