**PKCS # 7: 加密消息语法（规范）**

1.5版

**现状**

本文为互联网社区提供信息。它不是特指任何网络的一个标准。分发使用不受（网络）限制。

**版权通知**

版权所有 互联网协会（1998）。保留所有权利

**概述**

本文描述的是一个通用的数据的语法，可以应用于密码加密，例如： 数字签名、数字信封。本语法规则允许递归，例如，信封可以嵌套在另一个信封中，也可以由一方签名以前的封装的数字数据。它还允许将其它属性，如签名时间，和消息的内容一起认证，并提供其它与签名相关联的属性如副署/会签。（还可以）使用简化语法来传递证书（链）和证书吊销列表。

**1.范围**

**2.引用**

**3.定义（名词）**

本文档中将用到以下名词定义：

AlgorithmIdenifier 算法标识OID： 一个类型， 包括一个算法和关联参数的标识号。这个类型在X.509规范中定义。

ASN.1: 抽象语法符号1，在X.208规范中定义。

Attribute: 一个类型，包括一个属性类型（参数）和1~n个属性值。这个类型在X.501规范中定义。

BER: 基本编码规范，见X.209规范。

Certificate: 证书类型 X.509

CertificateSerialNumber: 证书序列号

CertificateRevocationList: 证书吊销列表

DER: Distinguished Encoding Rules for ASN.1, 见X.509 第8.7节；ASN.1的区分编码规则。

DES: 数据加密标准，见FIPS PUB 46-1标准。

desCBC: DES的密文分组链接模式。见NIST91标准。

ExtendedCertificate: 扩展证书

MD2: MD2摘要算法

md2: MD2的算法标识

MD5: MD5摘要算法

md5: MD5的算法标识

Name: 一种类型， 一个在X.500词典中的 有唯一标识和区分对象的类型。它被定义在X.501标准。在X.509证书中，这个类型用来标识证书颁发者和公钥的持有者

PEM: 隐私增强邮件，见RFCs 1421, 1422, 1423, 1424标准。

RSA：RSA公钥加密体系。 见RSA78标准。

**4.符号和缩写**

无

**5.一般概述**

接下来的9个章节分别讲 有用的类型，一般语法，六种内容类型和对象标识符OID。

本语法一般足够支持多种不同的内容类型。本文定义六种：数据，签名数据，封装数据，签名-数字信封-数据，摘要数据和加密数据。未来可能会增加新的内容类型。内容类型的使用可以在本文件之外定义，但必须遵守双方交换内容的双边协议。

本文通过各种对象标识符（指定）导出一种类型的ContentInfo。

有两种类的内容类型：基本的和增强的。基本类的内容类型包括“仅数据”，它没有算法加密的增强。目前，这个类里只有一种内容类型，就是数据类型。增强类中内容类型包括一些类型的内容（可能加密的）和另外一些加密增强的内容。比如说，封装数据（enveloped-data）内容可以包含（加密的）签名数据内容，它（里面）含有数据内容。四种非数据内容类型都是增强类里面的类型。增强类中的内容类型以此采用封装，从而产生术语“外部”内容（包含增强的内容）和“内部”内容（正在增强的内容）。

文档设计的增强内容类型可以为不定长的BER编码的单通数据作的准备，和任何BER编码的单通传输过程。单通传输的操作在磁带上存储内容或为别的任何“管道”传输的都特别有用。然而，单通操作的缺点之一是很难在单通中输出DER编码，因为各种组件的长度可能事先不知道。由于签名数据、签名和信封数据以及摘要数据内容类型需要DER编码，因此当数据以外的内容类型是这些内容类型之一的内部内容时，可能需要额外的传递。

**6.有用的类型**

本章节定义哪些至少在本文被至少引用两处的类型。

6.1证书吊销列表 CertificateRevocationLists

CeritificatRevocationLists ::=

SET OF CertificateRevocationList

6.2 内容加密算法标识者 ContentEncryptionAlgorithmIdentifier

ContentEncryptionAlgorithmIdentifier ::=

AlgorithmIdentifier

6.3 摘要算法标识符DigestAlgorithmIdentifier

DigestAlgorithmIdentifier ::= AlgorithmIdentifier

6.4 摘要加密算法标识符DigestEncryptionAlgorithmIdentifier

DigestEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier

6.5 扩展证书或证书

ExtendedCertificateOrCertificate ::= CHOICE {

certificate Certificate, -- X.509

extendedCertificate [0] IMPLICIT ExtendedCertificate

}

6.6 扩展证书集或证书集

ExtendedCertificatesAndCertificates ::= SET OF ExtendedCertificateOrCertificate

6.7 颁发者和序列号 IssuerAndSerialNumber

IssuerAndSerialNumber ::= SEQUENCE {

issuer Name,

serialNumber CertificateSerialNumber

}

6.8 密钥加密算法标识符 KeyEncryptionAlgorithmIdentifier

KeyEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier

6.9 版本 Version

Vesion :: = INTEGER

**7.一般语法**

本文表述的根据内容类型关联的数据内容的用户交换内容的语法。该语法使用ASN.1结构表示类型ContentInfo

ContentInfo ::= SEQUENCE {

contentType ContentType,

content [0] EXPLICIT ANY DEFINED BY contentType OPTIONAL

}

contentType: 1.2.840.113549.1.7.1~6六种类型 见8、10、11、12、13、14

以下各个数据域的含义：

1. contentType表示内容的类型。它是一个对象标识符，使用管理部门（IETF/RFC）分配的唯一的数字字符串来定义数据内容的类型。本文定义6中内容类型（参见第14章）：data, signedData, envelopedData, signedAndEnvelopedData, digestedData, and encryptedData。
2. content 是内容。这个数据域是可选的，如果这个字段不存在，它的值就必须以其它方式提供。这个类型的结构是由contentType属性指定定义的。

注意：

1. 由于内容的结构是由内容类型唯一声明的，所以对象标识符定义的类型不能是一个CHOICE类型。
2. 当ContentInfo的值是signed-data, signed-and-enveloped-data, 或digested-data 中的一种，那么消息摘要算法应用于DER编码的八进制内容。而ContentInfo的值是enveloped-data或signed-and-enveloped-data，那么内容加密算法应用于定长BER编码的八进制内容。
3. 省略内容中的可选域，可以构造出“外部签名”，例如，不修改或复制签名内容。外部签名的事例中，被签名的内容将从ContentInfo的内部封装中省略

**8.数据内容类型**

这种纯数据内容类型，就是仅仅用一个Octet string来表示。它必须有一个ASN.1类型结构的数据。

Data :: = OCTET STRING

这种数据内容类型意在关联（存储）任意八进制字符，比如ASNII文本文件；这些数据（存什么数据）完全由应用方来定。这些字符串可以是不需要任何结构，当然它也可以是另一个DER编码结构的数据。

**9.签名数据内容类型**

签名-数据内容类型， 由任何类型的内容和0至多个签名者对内容数据的加密信息摘要（签名）组成。这个签名者的加密摘要信息就叫作签名者对数据内容的“数字签名”。任何类型的内容都要被多个签名者并行地签名。此外，该语法还有个没有签名者的简化应用。这个简化应用用于传递（传播）证书和证书吊销列表。

一个签名-数据内容类型的典型应用是表示一个签名者对数据内容的数字签名。另一个典型的应用是用于分发证书和证书吊销列表。

签名数据的组装过程涉及以下几步：

1. 对于每个签名者，消息摘要的计算取决于签名者各自指定的摘要算法。（如果两个签名者使用相同的消息摘要算法，那么摘要信息只需要计算它们中的一个。）如果签名者验证的是除内容以外的其它信息（请参阅第9.2节），则签名者将使用摘要算法对内容和其他信息进行摘要，并且结果将作为“消息摘要”（，而非“数字签名”）。
2. 对于每个签名者，摘要信息和关联信息是各自使用自己的私钥加密的。
3. 对于每个签名者，加密信息摘要和其它签名者指定的信息将一起被放在SignerInfo中（请参阅第9.2节）。签名者的证书（链）和吊销列表，和其他无关签名者的信息将在这一步组装。
4. 所有签名者的消息摘要算法和签名数据将被合在一起，组装到SignedData中（请参见第9.1节）。

接收者通过签名者的公钥解密加密信息摘要得到内容的摘要，并与内容使用签名者指定摘要算法单独计算的摘要进行比对，来验证信息。签名者的公钥可以放在证书中，也可以由相关联的颁发者和颁发者中指定的序列号唯一确定的证书来获得公钥。

本章节分为5个部分。第一部分描述顶级签名-数据类型，第二部分每个签名者信息的类型SignerInfo,第三和第四部分描述消息摘要和摘要加密的过程，第五部分简述和PEM的兼容。

**9.1 SignedData 类型**

SignedData类型是一个ASN.1类型的结构：

SignedData ::= SEQUENCE {

version Version,

digestAlgorithms DigestAlgorithmIdentifiers,

contentInfo ContentInfo,

certificates [0] IMPLICIT ExtendedCertificatesAndCertificates OPTIONAL,

crls [1] IMPLICIT CertificateRevocationLists OPTIONAL,

signerInfos SignerInfos

}

DigestAlgorithmIdentifiers ::=

SET OF DigestAlgorithmIdentifier

SignerInfos ::= SET OF SignerInfo

signedData 类型内容各域如下：

1. version 语法的版本号，本文为1
2. digestAlgorithms 摘要算法标识的集合，包含消息摘要的算法标识符。它的大小任意，可以为零（当然不能是负数）。集合中每个元素标识一种消息摘要算法，用于指定下面的签名者使用哪种摘要。该集合旨在以任务顺序列出签名者的摘要算法，以便一次验证签名值。消息摘要的步骤描述参见第9.3节。
3. contentInfo是待签名的数据内容，它可以是任何定义了的内容类型。
4. certificates 是证书（链）集合，是一个符合PKCS#6扩展证书和X509证书的集合。它意在使用一个集合，足以把SignerInfos中的所有签名者的证书链包括根证书和其它顶级证书颁发机构包含在内。可能有多余的证书，甚至包含两个或多个来自不同的、独立的顶级根证书颁发机构的证书链。但也可以，如果签名者有别的方法获得证书，例如从前面的证书链中获得，集合也可以少于必需的数量。
5. crls 是一个证书吊销列表的集合。它意在通过集合足够将不在“热列表”的证书传播出去，但通讯不是必要的。证书吊销列表可能比预期多，也可能比预期少。
6. signerInfos是每个单个签名者信息的集合。集合数量不限，也可以是零。

注意：

1. 实际上，digestAlogrithms域在contentInfo和signerInfos域之前，这就使得它有可能一次通过。
2. 版本1与版本0的区别（版本0定义在pkcs#7, 1.4版，本版为1.5）如下：
   1. digestAlgorithms 和 singerInfos域在版本1中允许空集，而版本0中不允许。
   2. crls域在版本1中支持，而版本0中不支持。

解释下版本号的不同，版本0的签名值是可以被版本1兼容的。因此，按版本1实现来，可以处理两个版本的signedData。建议PKCS实现只生成版本1的SignedData值，但要准备好处理两个版本的SignedData值。

1. 通常，没有签名者的内容，ContentInfo值被签名是不紧要的。在这种情况下，建议将ContentInfo值的内容类型“signed”为data，并省略ContentInfo值的内容字段。

**9.2 SignerInfo 类型**

单个签名者的信息结构：

SignerInfo ::= SEQUENCE {

version Version,

issuerAndSerialNumber IssuerAndSerialNumber,

digestAlgorithm DigestAlgorithmIdentifier,

authenticatedAttributes [0] IMPLICIT Attributes OPTIONAL,

digestEncryptionAlgorithm DigestEncryptionAlgorithmIdentifier,

encryptedDigest EncryptedDigest,

unauthenticatedAttributes [1] IMPLICIT Attributes OPTIONAL

}

加密摘要/签名

EncryptedDigest ::= OCTET STRING

签名者信息的属性定义如下：

|  |  |  |
| --- | --- | --- |
| SignerInfo  签名者信息 | version  版本 | 语法标准的版本，本文为1 |
| issuerAndSerialNumber  颁发者和序列号 | 指定签名者证书的颁发机构和颁发机构指定的（证书）序列号 |
| digestAlgorithm  摘要算法 | 摘要算法的标识，内容数据和（如果有）认证属性，用该算法标识的摘要算法来做摘要处理。消息摘要过程见第9.3节 |
| authenticatedAttributes  已验证的属性 | 是一个签名者签名的属性集合。该项是可选的，但如果被签名的Content的内容类型不是数据，它就必须存在。如果它存在，那么它至少要包含两个属性：   1. 一个PKCS#9内容类型属性，表示被签名的ContentInfo值的内容类型； 2. 一个PKCS#9 消息摘要属性，表示内容的摘要的值。   还有其它可能有用的属性也可放在这里，比如签名时间，但必须是用PKCS#9定义的。 |
| digestEncryptionAlgorithm  摘要加密算法 | 摘要加密算法标识，标识一个签名者对消息摘要及相关加密信息的私钥加密算法。摘要加密过程详见第9.4节。 |
| encryptedDigest  加密摘要/数字签名 | 它是签名者对消息摘要及相关信息的私钥加密结果（值）。 |
| unauthenticatedAttributes  非认证属性 | 它是没有被签名者签名的一系列属性的集合，它是可选的。有些有用属性可以放在这儿，比如副署（会签），同时它必须使用PKCS#来定义。 |

注意：

1. 为了与PEM兼容，无论什么类型的ContentInfo的内容被签名，已验证属性都会被忽略，并且没有未验证属性项。
2. 版本1与版本0的不同之处是消息摘要的加密过程，详见第9.4节。PEM兼容时不同，反应了私有增强邮件的签名方面；而非PEM兼容下，消息摘要加密过程是没有差异的。建议PKCS实现只生成版本1的SignedData值，当版本0的PEM签名方法过时，建议使用版本1的PKCS实现来接收签名值。

**9.3 消息摘要过程**

消息摘要过程的计算是对待签名内容，或者待签名内容和签名者要认证属性一起的一个消息摘要。无论是哪种情况，待签名内容都是作为消息摘要过程的初始值。特别指出，这个“值”是ContentInfo项中的contents八进制位编码数据，为签名过程所应用的。只有内容的八进制位编码将被摘要，而不包括它的标识符和长度。

消息摘要的结果（我们叫作“消息摘要”）明显取决于要认证属性是否存在。当要认证属性缺少时，“消息摘要”即仅仅为对内容的摘要结果。当要认证属性存在时，那么“消息摘要”的结果将是对完整的要认证属性的编码数据的消息摘要。（显然：要认证属性中的隐式[0]标签不是Attributes的一部分。Attributes的标签是SET OF，而不是IMPLICIT[0]，SET OF的DER编码数据将和长度标签一起进行摘要。）当attributes存在时，那么它必须包含内容类型和对内容的消息摘要，这样“值”就间接包含在结果（“消息摘要”）中。

当要认证属性缺少时以及待签名数据含有内容类型时，仅仅只有内容的值被摘要。这样做的优势是不需要在消息加密过程之前知道要待签数据的长度。这个方式可兼容私有增强邮件PEM。

虽然八进制标识符和长度没有被摘要，但他们依旧被这个方式给保护了。之所以说长度也被保护了，是因为摘要算法是在假设通过计算不可能（或难度极大）找到两个不同的消息获得相同长度的消息摘要。显然，内容的类型唯一确定了标识符的八进制，标识符被以下两种方式隐式保护了：一是要认证属性中包含了内容类型，二是是使用PEM兼容代替（见第9.4节）来应用，内容类型作为数据。

注意：其实消息摘要是计算DER编码的部分，并不是说DER编码是必须的（对数据的包装转换）。确切地说，对本文档标识的实现可以保存其它DER编码的对象，但是这些（实现）方式不应影响消息摘要的计算。

**9.4 摘要加密过程**

摘要加密过程的入参包括消息摘要的最终结果（“消息摘要”）和摘要算法标识符（或对象标识符）。摘要加密过程的结果是使用签名者的私钥对DigestInfo(BER编码)的加密结果。

DigestInfo ::= SEQUENCE {

digestAlgorithm DigestAlgorithmIdentifier,

digest Digest

}

Digest ::= OCTET STRING

DigestInfo 的域有以下含义：

1. 摘要算法标识符是内容和要认证属性的摘要，它必须和签名者信息里的摘要算法标识符一致。
2. 摘要是信息摘要步骤的结果。

注意：

1. 签名步骤的签名和pkcs#1中定义的签名的唯一不同是，p1的签名为符合X.509标准的签名规范使用BitString，而这里使用OctetString。
2. 加密步骤的入参通常是30或更小的八进制数。如果摘要加密算法是PKCS#1中的rsaEncryption，那么意味着加密的单块输入和RSA的模长也就是至少328比特位一样长，这是合理的，并且符合安全要求的。
3. 摘要算法标识符包含在摘要信息的值内，如果结果被破坏，那么摘要将不符合。假如，对手可以通过MD2摘要信息找到原文， 那么，对手就可能从签名信息找到有相同MD2摘要的原文，并通过原文做新的签名。只有签名者以前使用MD2时，此攻击才会成功，因为DigestInfo值包含消息摘要算法。如果签名者从不信任MD2算法并且总是使用MD5，那么MD2的危害不会影响签名者。但是，如果DigestInfo值只包含消息摘要，那么MD2的危害将影响使用任何消息摘要算法的签名者。

**9.5 与私有增强邮件兼容**

当ContentInfo的待签签名值类型是data，且没有要认证属性，消息摘要是算法使用md2或md5，摘要加密算法是pkcs#1中的rsa时，它与私有增强邮件中的MIC-ONLY 和MIC-CLEAR 过程类型兼容。所有条件中，消息摘要的加密产生过程符合PEM中的一种产生方式因为（以下）：

1. 当没有要认证属性时，输入值的消息摘要算法和PEM中的一样。在PEM中是MD2和MD5，和md2、md5一样。
2. 当没有要认证属性是，在PEM中签名者的私钥加密的值（尤其是在RFC1423中）和本文一样。PEM中的RSA私钥加密和PKCS#1中的rsa加密相同。

签名数据内容的其它部分（如证书链，吊销列表，算法标识符等）都很容易转换成PEM中的相关组成成员。

**10.封装数据内容类型**

封装数据内容类型由任何加密的数据和一至多个接收者的加密的内容加密密钥组成。一个加密的某个接收者的内容加密密钥和加密内容组成该接收者的“数字信封”。任意内容都可以被任意个接收者独立/并行的封装。

典型应用是，使用数字信封将数据内容，摘要数据内容或起那么数据内容传递给一或多个数字信封的接收者。

构建封装数据包括下列步骤：

1. 给原文加密的特定算法的内容加密密钥是通过随机数生成。
2. 对于每个接收者，再使用该接收者的公钥对内容加密密钥加密。
3. 对于每个接收者，密文和内容加密密钥的密文都被放在一个RecipientInfo中，详见第10.2节。
4. 内容使用内容密钥加密。（内容可能需要填充处理或者分成多个块大小，详见第10.3节）。
5. 所有接收者的RecipientInfo值都放到EnvelopedData的值中，详见第10.1节。

一个接收者使用自己的私钥解密内容加密密钥的密文得到加密密钥，再用加密密钥解密加密内容得到（原来的）内容。接收者的私钥取决于颁发者和序列号唯一标识的证书关联的公钥。

本章节分成四个部分。第一部分描述EnvelopedData的顶级类型，第二部分描述每个接收者RecipientInfo，第三和第四部分描述内容加密和密钥加密的过程步骤。

因为私有增强邮件总是包含数字签名，而不仅仅是数字信封，所以这种类型不与私有增强邮件兼容（虽然有些步骤会与PEM的副署相比较）。

**10.1 封装数据类型**

封装数据类型需为ASN.1类型的。

EnvelopedData ::= SEQUENCE {

version Version,

recipientInfos RecipientInfos,

encryptedContentInfo EncryptedContentInfo

}

RecipientInfos ::= SET OF RecipientInfo 见第10.2节

EncryptedContentInfo ::= SEQUENCE {

contentType ContentType,

contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier,

encryptedContent [0] IMPLICIT EncryptedContent OPTIONAL

}

EncryptedContent ::= OCTET STRING

EnvelopedData的属性描述如下：

1. version 是语法版本号。本文它必须为0.
2. recipientInfos是每个接收者信息的集合。该集合至少要有一个元素。
3. encryptedContentInfo是加密的内容信息。

RecipientInfos的属性描述如下：

1. contentType标识着内容的类型。
2. contentEncryptionAlgorithm标识着内容加密的算法（和任何相关参数），在其中加密内容。内容加密的步骤见第10.3节。这个算法所有接收者都是一样的。
3. encryptedContent是内容加密的结果。这项是可选的，如果它不存在，它的值需由其它方式提供。

注意，实际上接收者信息再加密内容信息之前可以使得封装过程为单通的。单通过程详见第5章。

**10.2 RecipientInfo类型**

RecipientInfo是单个接收者的信息表述：

RecipientInfo ::= SEQUENCE {

version Version,

issuerAndSerialNumber IssuerAndSerialNumber,

keyEncryptionAlgorithm KeyEncryptionAlgorithmIdentifier,

encryptedKey EncryptedKey

}

EncryptedKey ::= OCTET STRING

RecipientInfo各属性含义如下：

1. version是语法的版本，本文是0。
2. issuerAndSerialNumber指的是接收者证书（为的是接收者可辨识名和公钥）颁发者可辨识名和颁发者指定的证书序列号。
3. keyEncryptionAlgorithm 标识使用接收者公钥加密内容密钥的加密算法。加密密钥的过程详见第10.4节。
4. encryptedKey是使用接收者公钥加密内容密钥的结果（见下节）。

**10.3 内容加密过程**

待封装的数据“值”是内容加密过程的初始值。特别指出，这个值是一个定长的BER编码的八进制位内容，是ContentInfo的content成员。只有BER编码的八进制位会被加密，而不包括它的标识符和长度，就更别说其它的了。

当要封装的内容类型是data时，仅对data的值加密。不知道要加密的数据的长度是个优点。这个方式与PEM兼容。

八进制标识符和长度不被加密。在加密过程长度是否被隐藏保护取决于加密算法。假如数据类型被唯一指定位八进制的，那么即使（攻击者）可以从内容类型恢复内容标识符也没有意义，所以内容标识符就不需要保护了。在signed-and-enveloped-data内容类型的应用中，要求显式的保护内容标识符和长度。而在digest-data和enveloped-data内容中，则应用一串数据。

注意：

1. 使用定长BER编码的原因是在封装的过程中，任何地方都没有说明是使用定长还是不定长。定长编码更适合简单类型如OctetString，所以采用定长编码。
2. 一些内容加密算法要求被加密的数据被分为k长度的多个块，这就需要应用定义一个方法使初始值预处理成这样的多个块。这些算法将在初始值尾部接上一些数据，这数使用l mod k 来计算。换句话说：在初始值的尾部接收1-k个相同的数，像下面这样

01 – if l mod k = k -1

02 – if l mod k = k -2

.

.

.

k k … k – if l mod k = 0

补充的值可以在不破坏初始值的，也就是作为未补充值的后缀。这种填充的方法仅对k<256是好的定义；对于更大的k值的方法有待进一步研究。

**10.4 密钥加密过程**

密钥加密过程的初始值仅仅是内容加密的密钥，并使用接收者应用的加密算法和公钥进行加密。

**11.签名-数字信封-数据 内容类型**

**12.摘要数据 内容类型**

**13.加密数据 内容类型**

**14. 对象标识者（符）**