ICS 35.100.70
L 79

中华人民共和国国家标准化指导性技术文件

GB/Z 19717—2005

基于多用途互联网邮件扩展(MIME)的安全报文交换

Secure message interchange based on Multipurpose Internet Mail Extensions

2005-04-19 发布　　2005-10-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

前　言

本指导性技术文件主要参照 Internet 工程任务组提出的 RFC 2630 密码报文语法、RFC 2633 S/MIME 报文规范 第 3 版和 RFC 2634 增强的 S/MIME 安全服务制定的。

本指导性技术文件的附录 A 是资料性附录。

本指导性技术文件由中华人民共和国信息产业部提出。

本指导性技术文件由全国信息安全技术标准化技术委员会归口。

本指导性技术文件起草单位：中国电子技术标准化研究所。

本指导性技术文件主要起草人：吴志刚、赵菁华、王颜尊。

本指导性技术文件仅供参考。

引　言

Internet 的电子邮件在传输中广泛使用简单邮件传输协议(即 SMTP),而 SMTP 却未提供加密服务。攻击者可在邮件传输中截获数据,并能将邮件中的文本格式、非文本格式的二进制数据(如:.exe 文件)进行轻松地还原。Internet 电子邮件面临着各种安全威胁(如信息泄露、冒充身份等)。

安全电子邮件能够提供信息加密、身份鉴别、内容完整性、机密性及抗抵赖性等安全服务。目前,Internet 工程任务组研究制定的安全多用途互联网邮件扩展(S/MIME)规范已成为安全电子邮件的重要支撑标准。S/MIME 系列规范主要采用单向散列算法和公开密钥基础设施(PKI)来实现数据加密和数字签名,从而保证邮件的安全性。

本指导性技术文件给出了 S/MIME 系列规范的关键内容,便于对 S/MIME 系列规范的深入分析及相关产品的开发。

本指导性技术文件凡涉及密码相关内容,按国家有关法规实施。

本指导性技术文件中所引用的 MD5、SHA-1、DSS、RSA、DES、RC2、DH 密码算法等均为举例说明。

基于多用途互联网邮件扩展(MIME)的安全报文交换

1 范围

本指导性技术文件阐述了安全发送和接收多用途互联网邮件扩展(MIME)数据的基本方法(即安全多用途互联网邮件扩展,S/MIME)。该方法基于广泛使用的多用途互联网邮件扩展协议(MIME),向各种 Internet 报文应用提供鉴别、报文的完整性、抗抵赖性、机密性等多种安全服务。传统的邮件用户代理使用该方法可以向所发送的报文增加各种加密服务,并能够有效处理所收报文中的加密服务。本指导性技术文件还描述了 S/MIME 的增强安全服务。

本指导性技术文件不限于电子邮件,它还可以用于任何传输 MIME 数据的传输机制(如超文本传输协议,HTTP)。该规范利用了 MIME 面向对象的特点,使得在各种传输系统中能够交换安全报文。

2 规范性引用文件

下列文件中的条款通过本指导性技术文件的引用而成为本指导性技术文件的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本指导性技术文件,然而,鼓励根据本指导性技术文件达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本指导性技术文件。

RFC 2045 多用途 Internet 邮件扩展(MIME) 第 1 部分 Internet 报文体的格式

RFC 2630 密码报文语法

RFC 2633 S/MIME 报文规范 第 3 版

RFC 2634 增强的 S/MIME 安全服务

3 术语、定义和缩略语

3.1 术语和定义

下列术语和定义适用于本指导性技术文件。

3.1.1

证书 certificate

采用数字签名将实体的可辨别名与公开密钥捆绑起来的类型。

3.1.2

接收代理 receiving agent

一种软件,它解释并处理 S/MIME CMS 对象及含有 CMS 对象的 MIME 主体部分。

3.1.3

发送代理 sending agent

一种软件,它创建 S/MIME CMS 对象和创建含有 CMS 对象的 MIME 主体部分。

3.1.4

多用途互联网邮件扩展 Multipurpose Internet Mail Extensions(MIME)

MIME 容许以下格式文档作为报文:

a) 非 ASCII 码的字符集的文本报文体;

b) 非文本报文体的不同格式的扩展集;

c) 多部分的报文体；

d) 非 ASCII 码的字符集的文本头信息。

3.1.5

S/MIME 代理 S/MIME agent

一种用户软件，它可以是接收代理或发送代理，或两者都是。

3.2 缩略语

下列缩略语适用于本指导性技术文件。

CMS 密码报文语法

ESS 增强安全服务

MIME 多用途互联网邮件扩展

S/MIME 安全多用途互联网邮件扩展

4 密码报文语法(CMS)

4.1 概述

密码报文语法(CMS)是以数字方式签名、处理、鉴别或加解密任意报文的语法，并用来描述保护数据的封装语法。它支持数字签名、报文鉴别代码和加密。这种语法允许多重封装，即一个封装包可以嵌套另一个包。同样，一方可以用数字方式给一些先前已经封装过的数据签名。它还允许任意属性(如签名时间)同报文内容一起被签名，以及允许其它属性(如防范签名)与签名相关联。密码报文语法可以支持各种实现基于证书的密钥管理功能的体系结构。RFC 2630 对密码报文语法(CMS)有详细的规定。

密码报文语法(CMS)普遍支持许多不同的内容类型。RFC 2630 规定了一种保护内容：ContentInfo。ContentInfo 可封装一个已标识的内容类型，这个已标识的类型可以进一步进行封装。RFC 2630 规定了六种内容类型：数据、签名数据、包装数据、摘要数据、加密数据和鉴别数据。

4.2 密码报文语法基本结构

密码报文语法(CMS)将内容类型标识符与内容相关联起来。这种句法包含抽象语法记法 1(ASN.1)类型的 ContentInfo 字段：

```
ContentInfo ::= SEQUENCE {
  contentType ContentType,
  content [0]. EXPLICIT ANY DEFINED BY contentType }

ContentType ::= OBJECT IDENTIFIER
```

ContentInfo 字段有以下含义：

contentType 表示相关联内容的类型。它是一个对象标识符，是由定义内容类型的机构分配的唯一整数串。

content 是关联的内容。内容的类型可以由 contentType 唯一确定。

5 安全多用途互联网邮件扩展(S/MIME)

5.1 概述

安全多用途互联网邮件扩展(S/MIME)规定了向 MIME 数据增加加密签名和加密服务的方法。MIME 规范规定了 Internet 报文内容类型的通用结构，并对新的内容类型应用提供扩充机制。RFC 2633 则定义了如何按照 CMS 创建经加密的 MIME 主体部分，以及用于传输加密 MIME 主体部分的 application/pkcs7mime 内容类型。RFC 2633 还讨论了如何使用 MIME-SECURE 所定义的 multipart/signed 的 MIME 类型来传输 S/MIME 签名报文，同时还定义了另一种用于传输 S/MIME 签名报文的 application/pkcs7-signature 的 MIME 类型。为了创建 S/MIME 报文，S/MIME 代理必须遵守 RFC

2633 及 RFC 2630 中所列的其他规范。由于 S/MIME 系统可能涉及软件而不是传统的 Internet 邮件客户端，因此接收代理与发送代理的要求是不同的。任何传输 MIME 数据的系统都能采用 S/MIME，例如发送加密报文的自动进程可能完全不能接收加密的报文。

5.2 支持 S/MIME 的 CMS 选项

在内容和算法支持方面，CMS 考虑了各种各样的选项。为了使所有支持 S/MIME 版本 3 的实现方案间达到基本的互操作性，RFC 2633 提出了一些支持要求和建议。

5.2.1 摘要算法标识符

发送代理和接收代理必须支持同一个算法。这里以 SHA-1 为例。为了提供对采用 MD5 摘要的 S/MIME 版本 2 签名数据对象的向后兼容性，接收代理应支持 MD5。

5.2.2 签名算法标识符

发送代理和接收代理必须支持同一个数字签名算法，这里以 DSS 为例。发送代理和接收代理应支持在 DSS 中定义的 id-dsa，算法参数不必存在。接收代理应支持 PKCS-1 所定义的 rsaEncryption，发送代理应支持 rsaEncryption。采用用户私有密钥对出去的报文进行签名，并在密钥生成期间确定私有密钥的长度。

注：S/MIME 版本 2 客户只能够验证采用 rsaEncryption 算法的数字签名。

5.2.3 密钥加密算法标识符

发送代理和接收代理必须支持同一个密钥交换算法，这里以 Diffie-Hellman 算法为例。接收代理应支持 rsaEncryption。进入的加密报文包含着多个对称密钥，这些密钥可以通过用户的私有密钥来解密，在密钥生成期间确定私有密钥的长度。发送代理应支持 rsaEncryption。

注：S/MIME 版本 2 的客户只能解密使用 rsaEncryption 算法的内容加密密钥。

5.2.4 通用语法

CMS 规定了多种内容类型，其中 S/MIME 目前可用的只有数据、签名数据及包装的数据这三种内容类型。

5.2.4.1 数据内容类型

发送代理必须使用 id-data 内容类型标识符来指明那些已采用安全服务的报文内容。例如，当对 MIME 数据采用数字签名时，CMS signedData encapContentInfo eContentType 必须包括 id-data 对象标识符，并且 MIME 内容必须被存放在 SignedData encapContentInfo eContent 的八位位组串中（除非发送代理使用多部分/签字，此时 eContent 可以不存在）。另一个例子，当对 MIME 数据进行加密时，CMS EnvelopedData encryptedContentInfo ContentType 必须包含 id-data 对象标识符，并且加密的 MIME 内容必须存放在 envelopedData encryptedContentInfo encryptedContent 的八位位组串中。

5.2.4.2 签名数据内容类型

发送代理必须对应用数字签名的报文使用签名数据内容类型，或者在用于传输无签名信息的证书的退化情况下必须使用签名内容类型。

5.2.4.3 包装数据内容类型

该内容类型用来向报文提供隐私性保护。发送者必须能够获得应用该服务所涉及的每一报文接收者的公开密钥。

5.2.5 属性签名者信息类型

SignerInfo 类型允许同签名一起包含无签名属性和签名属性。接收代理必须能够处理在此列出的每个签名属性的零个或一个实例。发送代理应在每个 S/MIME 报文中生成下列每个签名属性中一个实例：

a) signingTime；

b) sMIMECapabilities；

c) sMIMEEncryptionKeyPreference 。

此外，接收代理应能处理 signingCertificate 属性内签名属性的零个或一个实例。发送代理应能在每个 S/MIME 报文中生成 signingCertificate 签名属性的一个实例。以后可能会对这些属性的附加属性和值进行定义。接收代理应能通过友好的方式来处理那些它不识别的属性或值。那些含有此处未列出的签名属性的发送代理应能向用户显示这些属性，以便用户知道被签名数据的所有属性。

5.2.6 SignerIdentifier SignerInfo 类型

S/MIME 版本 3 要求使用 SignerInfo 版本 1，即对于 SignerIdentifier 必须使用 issuerAndSerialNumber 的选项。

5.2.7 内容加密运算标识符

发送代理和接收代理必须支持采用同一数字加密算法。这里以 DES EDE3 CBC 为例进行加密和解密，以下简称“tripleDES”。接收代理应支持以 RC2 为例的兼容算法进行加密和解密。

5.3 创建 S/MIME 报文

S/MIME 报文是 MIME 主体和 CMS 对象的组合，可以使用几种 MIME 类型和几种 CMS 对象。被保护的数据总是规范的 MIME 实体，并将 MIME 实体和其他数据（如证书和算法标识符）交给产生 CMS 对象的 CMS 处理设施，CMS 对象最终被包装在 MIME 中。S/MIME 的增强安全服务规范（见 RFC 2634）提供了如何嵌套的实例及构造安全的 S/MIME 报文的方法。RFC 2634 提供了一个如何采用 multipart/signed 和签名的 application/pkcs7-mime 构成一个三重隐蔽包装的 S/MIME 报文的实例。S/MIME 为 enveloped-only 数据提供了一种格式，为 signed-only 数据提供了几种格式，并为签名及包装的数据提供了几种格式。

5.3.1 准备签名或包装用的 MIME 实体

S/MIME 被用于安全 MIME 实体。MIME 实体可能是 subpart、或是报文的 subparts、或是带有其所有子部分的整个报文。作为整个报文的 MIME 实体只能包括 MIME 头和 MIME 主体，而不包括 RFC 822 的头。注意，S/MIME 还能用于 Internet 邮件以外的应用所使用的安全 MIME 实体。本指导性技术文件描述的安全的 MIME 实体可认为是“内部”MIME 实体，即它可能是大的 MIME 报文中“最内部”的对象，在 5.3.2、5.3.4 和其他部分描述了将“外部”MIME 实体处理为 CMS 对象。MIME 规范给出了准备 MIME 实体的规程。在签名时可使用带有某些附加限制的相同规程。在此重复了 MIME 规范中的规程描述，本条还描述了一些附加的要求。

对于创建签名的、包装的或签名且包装的 MIME 实体应采用单个的规程。为了预防邮件传输期间可能出现的已知问题，推荐采用某些附加步骤，这些步骤对于采用 multipart/signed 格式的清晰签名尤为重要。建议对包装的报文或签名且封装的报文实施这些附加步骤，以保证该报文能够无修改地转发到任何环境。

第一步：依据本地的约定准备 MIME 实体。

第二步：将 MIME 实体的叶子部分转换为规范的形式。

第三步：对离开的 MIME 实体应用适当的传送编码。

当接收到一个 S/MIME 报文时，要处理对报文所附带的安全服务，该结果就是 MIME 实体。该 MIME 实体将传递给具备 MIME 处理能力的用户代理，由该代理将该 MIME 实体进行解码和表示后提交给用户或接收应用程序。有关准备签名或包装用的 MIME 实体的详细要求见 RFC 2633。

5.3.2 application/pkcs7-mime 类型

application/pkcs7-mime 类型被用来携带几种类型 envelopedData 和 signedData 的 CMS 对象。本条描述了 application/pkcs7-mime 类型的一般特性：如果 eContentType 是 id-data，携带的 CMS 对象通常包含了按 5.3.1 所描述方式准备的 MIME 实体；当 eContentType 包括了不同的值时，可以携带其它内容。

由于 CMS 对象是二进制数据，大多数情况下 64 基的传送编码是适合的，尤其是采用 SMTP 传输方式。所用的传送编码依赖于被发送对象的传输方式，它不是 MIME 类型的特征。

注意：本讨论涉及的 CMS 对象或“外部”MIME 实体的传送编码，完全不同于由 CMS 对象保护的 MIME 实体的传送编码，且与其无关。因为 application/pkcs7-mime 对象有几种类型，所以发送代理应该尽可能地帮助接收代理在无需对对象的 ASN.1 进行解码的条件下就能了解对象的内容。所有的 application/pkcs7-mime 对象的 MIME 头应该包括“smime-type”可选参数。

5.3.3 创建 Enveloped-only 报文

本条描述了对 MIME 实体只包装不签名的格式。值得注意的是，发送只包装不签名的报文不能提供数据完整性，这可能是通过经处理的报文将依然有效而可能改变了其意义的方式来替换密文。

第一步：按照 5.3.1 准备将要包装的 MIME 实体。

第二步：将 MIME 实体和其他所需的数据处理为 envelopedData 类型的 CMS 对象。除了为每个接收方加密 content-encryption 密钥的副本外，还应该为发起方加密内容加密密钥的副本并将该副本包含于 envelopedData 中。

第三步：将 CMS 对象插入到一个 application/pkcs7-mime MIME 实体中。

enveloped-only 报文的 smime-type 参数为“envelopeddata”，该类型报文的文件扩展是“.p7m”。

5.3.4 创建只有签名的报文

对于定义的 S/MIME 签名报文存在两种格式，即带有 SignedData 的 application/pkcs7-mime 和 multipart/signed。发送代理应首选 multipart/signed 格式，但接收代理应都能处理这两种格式。

5.3.4.1 选择 Signed-only 报文格式

当选择了特定的 signed-only 格式时，由于该格式取决于所有接收者的能力，同时该格式还取决于带有能验证该签名的 S/MIME 设施的接收者与不带有能观察该报文的 S/MIME 软件的接收者的相对重要性，因此不存在严格的规则。

不论接收者是否拥有 S/MIME 软件，该接收者都能观察到查看采用 multipart/signed 格式签名的报文。无论接收者正使用本地 MIME 用户代理或它们拥有由网关所转换的报文，这些报文还能被观察到。在该上下文中，“观察到”表示处理报文的能力，在实质上就像该报文不是一个签名报文，包含其他 MIME 结构。接收方不能观察到采用 signedData 格式签名的报文，除非接收方拥有 S/MIME 设施。但是，如果接收方拥有 S/MIME 设施，通常能够验证这些报文是否在传输中没被更改。

5.3.4.2 采用带有 SignedData 的 application/pkcs7-mime 的签名

这种签名格式使用 application/pkcs7-mime MIME 类型。创建该格式的步骤如下：

第一步：按照 5.3.1 条准备 MIME 实体。

第二步：将 MIME 实体和其他所需的数据处理为 signedData 类型的 CMS 对象。

第三步：将 CMS 对象插入到一个 application/pkcs7-mime MIME 实体中。

采用带有 SignedData 的 application/pkcs7-mime 报文的 smime-type 参数为“signed-data”，该类型报文的文件扩展是“.p7m”。

5.3.4.3 采用 multipart/signed 格式的签名

本格式是清晰签名格式。不带有任何 S/MIME 或 CMS 处理设施的接收方能够观察该报文。它使用了 multipart/signed MIME 类型。Multipart/signed MIME 类型有两部分。第一部分包含所签名的 MIME 实体；第二部分包括“分离签名”的 CMS SignedData 对象，该对象中不存在 encapContentInfo eContent 字段。

5.3.5 签名和加密

为了完成签名和加密，可以嵌套任何 signed-only 格式和 encrypted-only 格式。由于上述的格式全部是 MIME 实体，并且这些格式都能保护 MIME 实体，所以允许这种嵌套方式。S/MIME 实现方案必须能够在接收方计算机合理有限的资源内接收和处理任意嵌套的 S/MIME。或者首先对报文进行签名，或者首先对报文进行包装，由实施者和用户来决定选择何种方式。当首先进行签名时，签名人就通过包装被安全地隐藏起来了。当首先进行包装，签名人是暴露的，但无需去除包装就可能验证签名。

无论选择首先签名还是首先加密都存在安全分歧。对于先加密后签名报文的接收方能够确认加密块没被改变，但不能确定签名者和未加密报文内容之间的关系。对于先签名后加密报文的接收方能够假设签名的报文本身未被改变，但是某些细致的攻击者可能已更改了加密报文的未经鉴别的部分。

5.3.6 创建 Certificates-only 报文

Certificates-only 报文或 certificates-only MIME 实体用来传输证书（诸如对注册请求的响应），这种格式也能够用来传输证书撤销列表。

第一步：要使生成 signedData 类 CMS 对象的 CMS 生成进程能够使用证书，signedData encapContentInfo eContent 字段必须不存在，并且 signerInfos 字段必须为空。

第二步：将 CMS signedData 对象装入 application/pkcs7-mime MIME 实体中。

certs-only 报文的 smime-type 参数是“certs-only”。该类型报文的文件扩展是“.p7c”。

5.3.7 注册请求

对报文进行签名的发送代理必须拥有签名证书，以便接收代理能够验证签名。

5.3.8 标识 S/MIME 报文

因为 S/MIME 考虑到在非 MIME 环境中的互操作性，采用了几种不同的机制来携带类型信息，它成为标识 S/MIME 报文的一点点困难。表 1 列出了判断报文是否是 S/MIME 报文的准则。如果某报文符合下列要求，则它可认为是 S/MIME 报文。表 1 中的文件后缀取自内容类型头中的“名称”参数或内容安排头中“文件名”参数，给出文件后缀的这些参数未作为参数段的部分来列出。

表 1 S/MIME 报文判断准则

MIME 类型：application/pkcs7-mime 参数：any 文件后缀：any
MIME 类型：multipart/signed 参数：protocol="application/pkcs7-signature" 文件后缀：any
MIME 类型：application/octet-stream 参数：any 文件后缀：p7m，p7s，p7c

5.4 证书处理

接收代理必须提供某些证书检索机制，以便获得访问数字信封接收方证书。本文件不涉及 S/MIME 代理如何处理证书，而只是描述在证书确认后或拒收后该代理执行什么。对于最初的 S/MIME 推广应用，用户代理最低的要求是能够为期望的接收方自动生成报文，该接收方请求在签名的返回报文中的接收方证书。接收代理和发送代理也应提供一种允许用户为通信者“存储和保护”证书的机制，通过这种机制可以保证以后可以检索这些证书。

5.4.1 密钥对的生成

如果 S/MIME 代理需要生成一个密钥对，那么 S/MIME 代理或某些相关的管理性实用程序或功能必须能以用户的名义生成分开的 DH 和 DSA 公开密钥/私有密钥对。每一密钥对必须根据非确定性随机输入 RANDOM 的良好来源来生成，并且私有密钥必须以安全方式加以保护。

如果 S/MIME 代理需要生成一个密钥对，那么 S/MIME 代理或相关的管理性实用程序或功能应生成 RSA 密钥对。

6 S/MIME 的增强安全服务

6.1 概述

S/MIME 可提供下列四种可选的增强安全服务：

a） 签名收据；

b) 安全标签；

c) 安全邮件发送列表及管理；

d) 签名证书属性。

6.1.1 签名收据

报文始发者可以请求来自报文接收者的签名收据。通过把 receiptRequest 属性增加到请求收据的 SignerInfo 对象的 SignedAttributes 字段中来指出该请求。当请求这样做时，接收用户代理软件应自动地创建签名的证书，并按照邮件发送列表扩充选项，本地安全策略和配置选项返回该收据。返回签名收据给始发者提供了报文交付证明，并且使得始发者可向第三方表明接收者曾验证过初始报文的签名。通过签名已把该收据与初始报文捆绑起来；因此，仅当要签名报文时，才可以请求该服务。收据发送者也可以有选择地加密某一收据，以便在收据发送者和收据接收者之间提供机密性。

收据请求可以指出这些收据被发送给许多地点，不仅仅发送给该发送者(在事实上，收据请求可能指出这些收据不宜都发给该发送者)。为了验证收据，收据的接收者必须是初始报文的始发者或接收者。因此，该发送者应不请求将收据发送给没有准确的报文副本的任何人。

由于收据涉及双方的交互，收据这一术语有时可能存在混淆。因此，在本章中，“发送者”就是发送包含请求收据的初始报文的代理。“接收者”就是收到报文和生成收据的一方。

6.1.2 安全标签

安全标签是通过 S/MIME 封装来保护的有关敏感内容的安全信息的集合。“授权”是向用户授予访问某个对象的权利和/或特权的行为。“访问控制”是实施这些权限的手段。安全标签中的敏感信息可以与用户的权限相比较，以确定用户是否被允许访问通过 S/MIME 封装所保护的内容。安全标签可以用于其他用途，诸如路由选择信息的来源。这些标签通常描述了分等级的若干级别(“秘密的”，“机密的”，“受限制的”等等)或者这些标签是基于角色的，描述哪种人可以看到该信息(“患者”的保健小组，医疗账单代理，“不受限制的”等等)。

6.1.3 安全邮件发送列表及管理

发送代理必须为加密报文的每个接收者创建特定于接收者的数据结构。这个过程可能损害发送给大量接收者的报文性能。因此，通常要求邮件列表代理(MLA)对于每个接收者可以采用单个报文或执行特定于接收者的加密。

对报文始发者来说，MLA 似乎是常规的报文接收者，但是，MLA 担当邮件列表(ML)的报文扩充点。报文发送者将报文送给 MLA，然后，将报文再分发给 ML 的成员。这个过程卸下了各个用户代理按每个接收者的处理工作量，并且便于更有效管理大量 ML。ML 是受 MLA 服务的真实报文接收者，该 MLA 为邮件发送列表提供了密码服务和扩充服务。

除了对报文进行密码处理外，安全邮件发送列表还必须防止邮件循环。邮件循环是指，其中某一个邮件发送列表是第二个邮件发送列表的成员，而第二个邮件发送列表又是第一个邮件发送列表的成员。一个报文将按向各列表的所有其他成分发邮件的快速级联连续方式从某一个列表到达另一个列表。

为了防止邮件循环，MLA 使用了三重隐蔽包装报文的外部签名的 mlExpansionHistory 属性。mlExpansionHistory 属性在本质上是已经处理该报文的所有 MLA 的列表。如果 MLA 看到了在该列表中的它自己的唯一实体标识符，那么它就知道已经形成了一次循环，并且它不会再向该列发送报文。

6.1.4 签名证书属性

令人们担心的事是，要求 CMS SignedData 对象的签名者与 SignedData 对象的验证过程被捆绑起来的证书不是以密码方式与该签名自身捆绑起来。本条也提出了对一组可能攻击的描述，这些攻击涉及被替换的某一证书用来验证所要求的证书的签名。针对可能的签名验证过程至少可以发起三种不同的攻击，其手法是替换签名验证过程中所使用的一个证书或多个证书。

a) 第一种攻击涉及用某一证书替换另一证书。在这种攻击中，SignedInfo 中的证书签发者和序列号被修改，以便提供新的证书。在签名验证过程期间使用这个新的证书。这种攻击的第 1

个版本是一种简单的拒绝服务攻击，其中，无效证书替换了有效证书。当证书中的公开密钥不再与签名该报文所使用的私有密钥相匹配时，这就使该报文成为不可验证的。其第二个版本是用某个有效证书替换初始有效证书，其中，两个证书中的两个公开密钥相匹配。这样做允许根据与该报文的始发者潜在不同的证书限制来确认该签名。

b) 第二种攻击涉及重新颁发签名证书(或可能是其证书之一)的认证机构 CA。随着认证机构重新颁发它们自己的根证书，或随着认证机构在重新颁发它们的根证书的同时改变证书的策略，这种攻击可能开始变为更频繁。在验证签名的过程中使用交叉证书(带有可能的不同限制)时就会出现这个问题。

c) 第三种攻击涉及建立一个 CA 的虚假实体，而这个虚假实体企图重复现有 CA 的结构。特别是，该虚假实体使用与该签名者使用的公开密钥相同的公开密钥来颁发新证书，但该虚假实体却使用其私有密钥对新证书进行签名。

为了防止或指出对应这些攻击的一组方法，以便处理一些最简单的攻击：

a) 对替换攻击的响应：

不能防止拒绝服务攻击。在运输中已经修改证书标识符之后，就不可能验证该签名。因为不能辨别被损坏的报文，也就没有任何自动标识出这种攻击的方法。对有效证书的替换可以用两种不同的方式进行响应。第一种方式是作出一个一般性声明，该声明指出在两个不同的证书中使用相同公开密钥是不良习惯并且必须避免这样做。实际上，没有实用的方法可防止用户获及带有相同公开密钥的新证书，而应该假设用户会这样做。将新属性包含在 Signer Info 签名属性中，这样做可以把正确的证书标识符捆绑到该签名中去。这样就将一个可能会成功的攻击转换为简单的拒绝服务攻击。

b) 对重新签发证书的响应：

认证机构决不应重新颁发一个带有不同属性的证书。这样的认证机构是跟随不良习惯，并且是不可信赖的。使用散列证书作为证书的基准可以防止针对端实体证书的这种攻击。为了防止基于重新颁发 CA 证书的攻击，可以要求对提出的 SigningCertificate 属性的用法有相当大的变化。要求将 ESSCertID 包含在该证书中，以表示在签名者的认证通路中的该颁发者的证书。当信赖的一方使用交叉证书作为其鉴别过程的一部分时，问题就出现了，同时该证书并不出现在证书列表上。在封闭 PKI 之外的这些问题使得系统添加这种信息时易于出错，并可能导致有效证书链被拒绝。

c) 对欺诈复制 CA 的响应：

防止这种攻击的最好方法是避免信任虚假 CA。使用散列来标识证书可防止使用来自虚假机构的端实体证书。然而，防止这种攻击的唯一实际方法是决不信任虚假 CA。

授权信息可以用作签名过程的一部分。该信息可以被携带任一属性证书和其他公开密钥证书中。签名者需要具有限制证书集合用于签名过程的能力，并且对信息需要进行编码，以使对 SignedData 对象的签名可包括该信息。本条中的方法便于将授权证书集合作为签名证书属性的一部分加以列出。

明确的证书策略也可以用作签名验证过程的一部分。如果签名者要求说明在确认该签名时宜使用的明确的证书策略，则该策略需要以密码方式捆绑到该签名过程中去。本条所描述的方法便于将证书策略声明集合作为签名证书属性的一部分加以列出。

6.2 三重隐蔽包装

6.2.1 基本概念

每项 S/MIME 增强安全服务的一些特性都采用“三重隐蔽包装”的概念对报文进行包装。即一个被三重隐蔽包装的报文要先被签名，然后被加密，最后又被签名。这里，对内部和外部签名的人可能是不同的实体或是同一个实体。

6.2.2 三重隐蔽包装的目的

并非所有的报文都需要三重隐蔽包装。当必须对报文签名、然后再加密，最后将签了名的报文属性装配到被加密的主体上时，需要使用三重隐蔽包装。发出报文的人或中间代理可以添加或删除其外部属性，且外部属性也可以被中间代理或最终的收件人签名。

内部签名可用于保证内容的完整性、抗抵赖、对信源的鉴别以及将报文的属性（如安全标签）捆绑到报文的初始内容中。这些属性从发件人传到收件人，无论中间实体（如处理报文的邮件清单代理）有多少个。这种签名属性可以用来控制对内部主体的访问，并且在内部签名中还带有发方索要签名收据的请求。

被加密的主体具有机密性，包括在内部签名中所携带的这些属性的机密性。

外部签名可用于鉴别被逐级处理的信息并可保证其完整性。每一级是一个中间实体，如邮件列表代理。外部签名将属性（如安全标签）捆绑到被加密的主体上，这些属性常用于访问控制和路由选择。

6.2.3 三重隐蔽包装的步骤

以下是创建一个三重隐蔽包装报文的步骤：

a) 先从报文主体开始，称“初始内容”。

b) 用适当的 MIME 内容类型（Content-type）头封装初始内容，例如“Content-type：text/plain”（内容类型：文本/明文）。此 MIME 封装规则的一个例外是被签名的收据不放在 MIME 头中。

c) 给第 2 步的结果（内部 MIME 头和初始内容）签名。SignedData encapContentInfo eContentType 对象标识符必须是 id-data。如果您在第 4 步创建的结构是 multipart/signed，则不得有 SignedData encapContentInfo eContent。如果您在第 4 步创建的结构是 application/pkcs7-mime，则 SignedData encapContentInfo eContent 必须包含上述第 2 步的结果。SignedData 的结构被内容类型（contentType）为 id-signedData 的 ContentInfo 序列所封装。

d) 按照 MSG [MSG]中的定义，将适当的 MIME 结构添加到第 3 步的被签名的报文中。所得出的报文称为“内部签名”。

——如果您用 multipart/signed 进行签名，则添加的 MIME 结构包括带参数的内容类型 multipart/signed、边界、上述第 2 步的结果、边界、内容类型 application/pkcs7-signature、可选的 MIME 头（如 Content-transfer-encoding 和 Content-disposition）以及作为上述第 3 步的结果的主体部分。

——如果您使用 application/pkcs7-mime 进行签名，则所添加的 MIME 结构包括带参数的内容类型 application/pkcs7-mime、可选的 MIME 头（如 Content-transfer-encoding 和 Content-disposition）以及上述第 3 步的结果。

e) 将第 4 步的结果作为单个块进行加密，将它变成一个 application/pkcs7-mime 对象。EnvelopedData encryptedContentInfo 的内容类型（contentType）必须是 id-data。EnvelopedData 结构被内容类型为 id-envelopedData 的 ContentInfo 序列所封装，并称为“被加密的主体”。

f) 添加适当的 MIME 头：带参数的内容类型 application/pkcs7-mime 和可选的 MIME 头部，如 Content-transfer-encoding 和 Content-disposition。

g) 用与上述第 3 步相同的逻辑，将第 6 步的结果（MIME 标题和被加密的主体）作为单个块进行签名。

h) 用与上述第 4 步相同的逻辑，将适当的 MIME 结构添加到第 7 步的被签名的报文中。所得到的结果称为“外部签名”，且是三重隐蔽包装的报文。

6.2.4 三重隐蔽包装报文的格式

一个被三重隐蔽包装的报文具有多层的封装。其结构会因报文中被签名的部分的格式的不同而不同。由于 MIME 封装数据的方式，这些封装层不按次序出现，使得“层”的概念变得模糊起来。

由于已知接受方能够处理 S/MIME 报文（因为他们解密了中间伪装器），因此无需在内部签名中使

用 multipart/signed 格式。发送代理了解选择在外层使用 multipart/signed 格式，这样，非 S/MIME 代理可以知道下一个内层被加密；然而，这没有什么价值，这是因为收方可以发现报文的其余部分是无法读的。由于许多发送代理通常使用 multipart/signed 结构，因此所有接收代理必须能够解释 multipart/signed 或 application/pkcs7-mime 签名结构。

采用这两种签名用的 multipart/signed 的三重隐蔽包装报文的格式是：

[第八步] Content-type：multipart/signed；
[第八步]　protocol＝"application/pkcs7-signature"；
[第八步]　boundary＝outerboundary
[第八步]
[第八步] ——outerboundary
[第六步] Content-type：application/pkcs7-mime；　　)
[第六步]　smime-type＝enveloped-data　　)
[第六步]　　)
[第四步] Content-type：multipart/signed；　　|)
[第四步]　protocol＝"application/pkcs7-signature"；　　|)
[第四步]　boundary＝innerboundary　　|)
[第四步]　　|)
[第四步] ——innerboundary　　|)
[第二步] Content-type：text/plain　　% |)
[第二步]　　% |)
[第一步] Original content　　% |)
[第四步]　　|)
[第四步] ——innerboundary　　|)
[第四步] Content-type：application/pkcs7-signature　　|)
[第四步]　　|)
[第三步] inner SignedData block（eContent is missing）　　|)
[第四步]　　|)
[第四步] ——innerboundary——　　|)
[第八步]
[第八步] ——outerboundary
[第八步] Content-type：application/pkcs7-signature
[第八步]
[第七步] outer SignedData block（eContent is missing）
[第八步]
[第八步] ——outerboundary——

这里：

% ＝ 这些行表示计算内部签名。

| ＝ 这些行表示在第 5 步被加密。加密的结果是不透明的，是 EnvelopedData 块的一部分。

) ＝ 这些行表示计算外部签名。

采用这两种签名用的 application/pkcs7-mime 的三重隐蔽包装报文的格式是：

[第八步] Content-type：application/pkcs7-mime；
[第八步]　smime-type＝signed-data
[第八步]

[第七步] outer SignedData block (eContent is present) O
[第六步] Content-type: application/pkcs7-mime;) O
[第六步] smime-type=enveloped-data;) O
[第六步]) O
[第四步] Content-type: application/pkcs7-mime; |) O
[第四步] smime-type=signed-data |) O
[第四步] |) O
[第三步] inner SignedData block (eContent is present) I |) O
[第二步] Content-type: text/plain I |) O
[第二步] I |) O
[第一步] Original content I |) O

这里：

I = 这些行是内部 SignedData 块，不仅是无法理解的，而且包含第 2 步中用 ASN.1 编码的结果以及控制信息。

| = 这些行表示在第 5 步被加密。被加密的结果是无法理解的，是 EnvelopedData 块的一部分。

) = 这些行表示计算外部签名。

O = 这些行是外部 SignedData 块，不仅无法理解，而且还包含第 6 步中用 ASN.1 编码的结果和控制信息。

6.3 S/MIME 增强安全服务和三重隐蔽包装

6.3.1 签名收据与三重隐蔽包装

在任何 SignedData 对象中，可能要求使用签名收据。如三重隐蔽包装的报文要求收件人回送收到该报文的签名收据给发件人时，其收据请求必须包含在内部签名中，而不能在外部签名中。因为，当邮件发送列表处理三重隐蔽包装报文时，安全邮件发送列表代理可能会改变该报文的外部签名中的收据策略。

6.3.2 安全标签与三重隐蔽包装

任何 SignedData 对象的签名属性可能包含安全标签。在内部签名、或在外部签名、或在这两种签名中都可能含有安全标签属性。

内部安全标签用来确定对与初始的明文内容有关的访问控制。内部签名提供鉴别，并以密码的方式保护位于内部主体的最初签名者的安全标签的完整性。这种策略便于转发报文，因为最初签名者的安全标签被包含在可以被转发给第三方的 SignedData 块中，使第三方可以验证包含内部安全标签的内部签名。机密性安全服务也可用于内部安全标签，方法是加密 EnvelopedData 块中的整个内部 SignedData 块。

外部 SignedData 块的签名属性也可包含安全标签，且该块可包含敏感的加密报文。外部安全标签用来确定与加密报文有关的访问控制和路由选择。

注意：安全标签属性只可以在 signedAttributes 块中使用。在 EnvelopedData 或未签名属性中不得使用 eSSSecurityLabel 属性。

6.3.3 安全邮件发送列表与三重隐蔽包装

安全邮件列表报文处理取决于发送给邮件列表代理的报文中所呈现的 S/MIME 各层的结构。如果存在内部签名，邮件列表代理决不会改变已散列(hashed)的数据，来形成内部签名。如果存在外部签名，邮件列表代理将修改已散列的数据，来形成这个外部签名。在这些情况下，邮件列表代理添加或更新 mlExpansionHistory 属性，并记录下邮件列表代理的处理活动，最终添加或替换待分发报文上的外部签名。

附 录 A
（资料性附录）
用 ASN.1 描述的语法定义

```
ExtendedSecurityServices
      { iso(1) member-body(2) us(840) rsadsi(113549)
        pkcs(1) pkcs-9(9) smime(16) modules(0) ess(2) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS

——密码报文语法（CMS）
    CryptographicMessageSyntax { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9
    (9) smime(16) modules(0) cms(1)}中的 ContentType, IssuerAndSerialNumber, ubjectKeyIden-
    tifier
——PKIX 证书和 CRL 框架 Sec A.2 隐藏标记模式，
——1988 语法
    PKIX1Implicit88 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms
    (5) pkix(7)id-mod(0) id-pkix1-implicit-88(2)} 中的 PolicyInformation
——X.509
    CertificateExtensions{joint-iso-ccitt ds(5) module(1) certificateExtensions(26) 0}中的 General-
    Names, CertificateSerialNumber;
——扩展安全服务
——在本模块中,“SEQUENCE SIZE (1..MAX) OF”结构出现在几个 ASN.1 结构中。一个有效的
    ASN.1 SEQUENCE 可以是零或更多项。SIZE (1..MAX) 结构将 SEQUENCE 限定为至少有
    1 项。MAX 表示没有规定上限。
——具体实现可以自由选择适合各自环境的上限。

UTF8String ::= [UNIVERSAL 12] IMPLICIT OCTET STRING
—— 在 [UTF8] 中介绍内容的格式。
——收据请求语法(Receipt Request Syntax)
ReceiptRequest ::= SEQUENCE {
  signedContentIdentifier ContentIdentifier,
  receiptsFrom ReceiptsFrom,
  receiptsTo SEQUENCE SIZE (1..ub-receiptsTo)) OF GeneralNames }

ub-receiptsTo INTEGER ::= 16

id-aa-receiptRequest OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 1}
```

```
ContentIdentifier ::= OCTET STRING

id-aa-contentIdentifier OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 7}

   ReceiptsFrom ::= CHOICE {
     allOrFirstTier [0] AllOrFirstTier,
     -- formerly "allOrNone [0]AllOrNone"
     receiptList [1] SEQUENCE OF GeneralNames }

   AllOrFirstTier ::= INTEGER { -- Formerly AllOrNone
     allReceipts (0),
     firstTierRecipients (1) }
```

——收据语法(Receipt Syntax)

```
Receipt ::= SEQUENCE {
  version ESSVersion,
  contentType ContentType,
  signedContentIdentifier ContentIdentifier,
  originatorSignatureValue OCTET STRING }

id-ct-receipt OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
   rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-ct(1) 1}

ESSVersion ::= INTEGER   { v1(1) }
```

——内容线索(Content Hints)

```
ContentHints ::= SEQUENCE {
  contentDescription UTF8String (SIZE (1..MAX)) OPTIONAL,
  contentType ContentType }

id-aa-contentHint OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
    rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 4}
```

——报文签名摘要属性(Message Signature Digest Attribute)

```
msgSigDigest ::= OCTET STRING
```

```
id-aa-msgSigDigest OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 5}
```

——签字内容参考属性(Signed Content Reference Attribute)

```
ContentReference ::= SEQUENCE {
  contentType ContentType,
  signedContentIdentifier ContentIdentifier,
  originatorSignatureValue OCTET STRING }

id-aa-contentReference    OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 10 }
```

——eSSSecurityLabel 语法(Syntax of eSSSecurityLabel)

```
ESSSecurityLabel ::= SET {
  security-policy-identifier SecurityPolicyIdentifier,
  security-classification SecurityClassification OPTIONAL,
  privacy-mark ESSPrivacyMark OPTIONAL,
  security-categories SecurityCategories OPTIONAL }

id-aa-securityLabel OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 2}

SecurityPolicyIdentifier ::= OBJECT IDENTIFIER

SecurityClassification ::= INTEGER {
  unmarked (0),
  unclassified (1),
  restricted (2),
  confidential (3),
  secret (4),
  top-secret (5) } (0..ub-integer-options)

ub-integer-options INTEGER ::= 256

ESSPrivacyMark ::= CHOICE {
    pString        PrintableString (SIZE (1..ub-privacy-mark-length)),
    utf8String     UTF8String (SIZE (1..MAX))
}
```

```
ub-privacy-mark-length INTEGER ::= 128

SecurityCategories ::= SET SIZE (1..ub-security-categories) OF
        SecurityCategory

ub-security-categories INTEGER ::= 64

SecurityCategory ::= SEQUENCE {
  type   [0] OBJECT IDENTIFIER,
  value  [1] ANY DEFINED BY type -- defined by type
```

注：上述 SecurityCategory 句法生成的十六位编码与 X.411 规范提供的以下 SecurityCategory 句法所生成的十六进制编码相同：

```
--
--SecurityCategory ::= SEQUENCE {
--       type   [0]  SECURITY-CATEGORY,
--       value  [1]  ANY DEFINED BY type }
--
--SECURITY-CATEGORY MACRO ::=
--BEGIN
--TYPE NOTATION ::= type | empty
--VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)
--END
```

——安全标签的等价物(Equivalent Security Labels)

```
EquivalentLabels ::= SEQUENCE OF ESSSecurityLabel

id-aa-equivalentLabels OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 9}
```

——邮件列表扩展历史语法(Mail List Expansion History Syntax)

```
MLExpansionHistory ::= SEQUENCE
        SIZE (1..ub-ml-expansion-history) OF MLData

id-aa-mlExpandHistory OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 3}

ub-ml-expansion-history INTEGER ::= 64
```

```
MLData ::= SEQUENCE {
  mailListIdentifier EntityIdentifier,
  expansionTime GeneralizedTime,
  mlReceiptPolicy MLReceiptPolicy OPTIONAL }

EntityIdentifier ::= CHOICE {
  issuerAndSerialNumber IssuerAndSerialNumber,
  subjectKeyIdentifier SubjectKeyIdentifier }

MLReceiptPolicy ::= CHOICE {
  none [0] NULL,
  insteadOf [1] SEQUENCE SIZE (1..MAX) OF GeneralNames,
  inAdditionTo [2] SEQUENCE SIZE (1..MAX) OF GeneralNames }
```

——签字证书属性定义(Signing Certificate Attribute Definition)

```
SigningCertificate ::=  SEQUENCE {
    certs          SEQUENCE OF ESSCertID,
    policies       SEQUENCE OF PolicyInformation OPTIONAL
}

id-aa-signingCertificate OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9)
    smime(16) id-aa(2) 12 }

ESSCertID ::=  SEQUENCE {
     certHash                 Hash,
     issuerSerial             IssuerSerial OPTIONAL
}

Hash ::= OCTET STRING -- SHA1 hash of entire certificate

IssuerSerial ::= SEQUENCE {
     issuer                   GeneralNames,
     serialNumber             CertificateSerialNumber
}

END -- of ExtendedSecurityServices
```

参 考 文 献

[1] [PKCS-7] Kaliski, B.,《PKCS #7：密码报文语法 1.5 版》RFC 2315，1998.3

[2] [PKCS-1] Kaliski, B.,《PKCS #1：RSA 加密算法 2.0 版》RFC 2437，1998.10

[3] [RANDOM] Eastlake, 3rd, D., Crocker, S. and J. Schiller,《用于安全的随机性建议》RFC 1750，1994.12

[4] [RC2] Rivest, R.,《RC2 加密运算描述》RFC 2268，1998.1

[5] [SHA1] NIST FIPS PUB 180-1,《安全 hash 标准》草案，1994.5.31

[6] [3DES] ANSI X9.52-1998,《三重数据加密运算操作模式》,1998

[7] [DES] ANSI X3.106,《美国国家信息系统标准-数据链接加密》,1983

[8] [DSS] FIPS Pub 186:《数字签名标准》,1994.5.19

[9] [MD5] Rivest, R.,《MD5 报文-运算法则摘要》，RFC 1321，1992.4