

RFC1113 ——Privacy Enhancement for Internet Electronic Mail:
Part I -- Message Encipherment and Authentication Procedures
Internet 电子邮件保密增强: Part1-消息编码和鉴别过程

组织: 中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者: 金凤 (phoenix_jin take.a.bow@263.net)

译文发布时间: 2001-10-28

版权: 本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载, 但必须保留本文档的翻译及版权信息。

Network Working Group

Request for Comments: 1113

Obsoletes RFCs: 989, 1040

J. Linn

DEC

IAB Privacy Task Force

August 1989

Internet 电子邮件保密增强: Part1-消息编码和鉴别过程

(RFC1113 ——Privacy Enhancement for Internet Electronic Mail:
Part I -- Message Encipherment and Authentication Procedures)

本备忘录的状态

本档讲述了一种 Internet 社区的 Internet 标准跟踪协议, 它需要进一步进行讨论和建议以得到改进。请参考最新版的“Internet 正式协议标准” (STD1)来获得本协议的标准化程度和状态。本备忘录的发布不受任何限制。

版权声明

Copyright (C) The Internet Society (2001).

感谢

本档是 internet 构架委员会 (IAB) 保密特别委员会的一系列会议及在那些会议上分发的 internet 工作报告的产物。我想感谢以下列出的保密特别委员会的成员集中了他们在会议上的观点和贡献形成了这个 rfc 文档: David Balenson, Curt Barker, Jim Bidzos, Matt Bishop, Danny Cohen, Tom Daniel, Charles Fox, Morrie Gasser, Russ Housley, Steve Kent (chairman), John Laws, Steve Lipner, Dan Nessett, Mike Padlipsky, Rob Shirey, Miles Smid, Steve Walker, and Steve Wilbur.

目录

1. 介绍	3
2. 术语	3
3. 服务、约束和暗示	3
4. 消息处理	5
4.1 消息处理总览	5
4.1.1 密钥类型	5

RFC1113 ——Privacy Enhancement for Internet Electronic Mail:
Part I -- Message Encipherment and Authentication Procedures
Internet 电子邮件保密增强: Part1-消息编码和鉴别过程

4.1.2 处理过程.....	6
4.2 加密算法和模式.....	6
4.3 保密增强消息转换.....	7
4.3.1 约束.....	7
4.3.2 建议.....	8
4.3.2.1 步骤一: 本地形式.....	8
4.3.2.2 步骤二: 规范形式.....	8
4.3.2.3 步骤三: 鉴别和加密.....	8
4.3.2.4 步骤四: 可打印的编码.....	9
4.3.2.5 转换概述.....	9
4.4 封装机制.....	10
4.5 邮件列表的邮件.....	12
4.6 被封装的头域小结.....	12
4.6.1 每个消息被封装的头域.....	14
4.6.1.1 X-Proc-Type 域.....	14
4.6.1.2 X-DEK-Info 域.....	14
4.6.2 一般每个消息被封装的头域.....	15
4.6.2.1 X-Sender-ID 域.....	15
4.6.2.2 X-Certificate 域.....	15
4.6.2.3 X-MIC-Info 域.....	15
4.6.3 不定出现的头域.....	16
4.6.3.1 X-Issuer-Certificate 域.....	16
4.6.4 每个接收者被封装的头域.....	16
4.6.4.1 X-Recipient-ID 域.....	16
4.6.4.2 X-Key-Info 域.....	17
4.6.4.2.1 对称密钥管理.....	17
4.6.4.2.2 非对称密钥管理.....	17
5. 密钥管理.....	17
5.1 数据加密密钥 (DEKs)	18
5.2 交互密钥 (IKs)	18
5.2.1 子域的定义.....	19
5.2.1.1 实体标识符子域.....	19
5.2.1.2 发行机构子域.....	19
5.2.1.3 版本/满期子域.....	19
5.2.2 IK 加密期发行	20
6. 用户命名.....	20
6.1 当前的方法.....	20
6.2 发行考虑.....	20
7. 用户接口和实现的例子.....	21
8. 进一步研究的领域.....	21
9. 参考.....	21
注意:	22
作者地址:	24

1. 介绍

本文档定义了为在 **internet** 上传输的电子邮件提供保密增强服务的消息编码和鉴别的过程。是四个相关文档中的一篇。在当前的 **RFC** 中定义的步骤试图与各种密钥管理方法保持兼容,包括加密密钥的数据加密的对称(密钥)和非对称(公钥)方法。并预见了消息文本加密的对称密码系统的使用和 / 或完整性检查计算。**RFC-1114** 规定了支持基于公钥证书使用的密钥管理机制。**RFC-1115** 规定了算法和与当前文档和 **RFC-1114** 相关的信息。后续的 **RFC** 将提供被建立的密钥基础设施的详细的报告和电子格式和过程以支持这些服务。

保密增强服务(机密性、可鉴别性、消息完整性保证)是通过发送者和接受者用户进程之间的端对端的加密系统提供的,没有特殊的处理要求强加于端点的消息传输系统或中继站。这种方法容许保密增强功能被结合到一个站对站或用户对用户的基础之上,不会影响其他的网络实体。支持不同种类的组件和邮件传输工具之间的互操作性。

2. 术语

为了描述的目的,本 **RFC** 文档使用了定义在 **OSI X.400** 消息处理系统(1984 年经 **CCITT** 推荐)模型里的术语。这部分复制了 **X.400 2.2.1** 小节的部分内容,“**MHS** 模型描述:总览”为了使不熟悉 **OSI MHS** 模型的读者清楚的理解这个术语。

在 **MHS** 模型中,一个用户是一个人或一个计算机应用。一个用户要么作为发送者要么作为接收者(当正在接收)。**MH** 服务元素定义了一组消息类型和一个发送者传送这些类型消息给一个或多个接收者的能力。

一个发送者在他或她的用户代理的帮助下准备消息。一个用户代理(**UA**)是一个和传送消息的消息传输系统(**MTS**)相互影响的应用进程。这个 **MTS** 向一个或多个接受者的用户代理传送消息。只是被用户代理操作而没有标准化为一个 **MH** 服务元素的一部分的功能被称为本地用户代理功能。

MTS 有大量的消息传送代理(**MTAs**)组成。在一起操作,**MTAs** 转送消息将它们传送给目的接收用户代理,通过这些代理使消息对于目的接受者可用。

UAs 和 **MTAs** 总称为消息处理系统(**MHS**)。**MHS** 和它的所有用户作为消息处理环境。

3. 服务、约束和暗示

本文档定义了增强电子邮件在网络上保密传送的机制。在本文档中讨论的功能提供了基于发送者和接收者用户代理的端对端的系统保密增强服务。没有保密增强被提供给通过中间节点转发和增加的消息域。

鉴别和完整性功能总是应用到整个消息文本,没有不通过鉴别的机密性功能,加密功能可以有选择的应用到消息的内容部分;这允许在接收者的个人密钥缺失的情况下

消息不太敏感的部分（例如，描述域）被接收者的代理处理。在有些情况下，消息整体被排除在加密之外，这一特色可以用于不含机密性的鉴别和完整性服务的有效组合。

为了和 internet 的不同支持者和使用模式保持一致，定义在文档中的各种方法可以应用到 internet 主机和使用范例的广泛的范围。特别下面的属性值得注意：

1. 定义在文档中的机制没有限制针对特殊的主机或操作系统，但是允许在大量系统间的互操作性。所有的保密增强在应用层实现，不依赖于底层协议的任何保密特色。
2. 被定义的机制和非增强的网络组件相兼容。保密增强在端到端的风格中中间转发的主机不影响邮件的处理，中间的主机没有加入保密增强功能。然而消息的发送者识别接收者是否实现保密增强，为了编码。可能加密不会被用于没有对应转换的消息的目的地。
3. 定义的机制是和一些的邮件传输功能相兼容的。在 Internet 内，电子邮件传输受各种 SMTP 的实现影响。一定的站点凭借 SMTP 可获取，传送邮件到其他的邮件处理环境（例如 USENET, CSNET, BITNET）。保密增强必须能通过 SMTP 领域操作；也要求与在 SMTP 环境和其他连接环境的电子邮件发送保护相一致。
4. 定义的机制和大范围的电子邮件用户代理相一致。各种各样的被用在 internet 电子邮件用户代理程序，和相应范围的用户接口范例。为了使电子邮件保密性增强最大可能的应用于用户交互，所选择的机制应该对于最大可能的各种存在 UA 程序可用。对于引导实现的目的，要求保密增强处理被组合成一个单独的程序，对于大多数的 UAs 可用，而不是去修改每一个提供保密增强服务的每个 UA。
5. 定义的机制允许电子邮件保密增强处理被操作在个人电脑上独立于不同的 UA 功能被实现的系统。给定 PCs 扩展的使用和被放置在许多多用户系统的 UA 实现的信任程度，这个属性能允许许多用户以比一个严格基于 UA 的方法所能允许保证级高的级别来处理保密增强邮件。
6. 定义的机制支持电子邮件定位的邮件列表保密保护（分发列表，ISO 用语）。
7. 定义在本文档中的机制和各种支持的密钥管理方法相兼容，包括（未限制）手工的预分发，基于对称密码的密钥分发，和公钥证书的使用。不同的密钥管理机制可以被用于一个广播消息的不同接收者。而为了与本文档的兼容性支持一个特殊的密钥管理机制不是最小的必要的要求，强烈推荐采用定义在 RFC-1114 中的公钥证书方法。

为了获得对于最大可能范围的网络主机和邮件系统的适用性，便利引导实现和测试而无须预先修改整个网络，在本文档中考虑了影响一组方法的三个基本的限制：

1. 方法将被限制于在端点的实现并将受用户代理级等完整性的影响，而不必集成到消息传输系统（例如，SMTP 服务）。
2. 被支持的方法增强而不是限制用户的能力。被信任的实现，包含完整性特色保护软件不被本地用户颠覆，不能做一般的假设。在这样的特色缺少的情况下，提供增强对用户服务的便利（例如，通过保护和鉴别中间用户交互）显然比增强对用户行为限制（内部用户获取控制）更加可行。
3. 被支持的一组方法集中于一组被选择提供广大用户社区重要的和切实的利益的功。通过集中最关键的服务组，我们旨在通过普通的实现努力最大化被增加的保密值。

由于这些限制，能提供下面的功能：

1. 解密保护，
2. 发送者鉴别，

3. 消息完整性方法,
4. (如果使用非对称密钥管理) 来源不可否认,
但是下面的与保密相关的影响未提到:
 1. 获取控制,
 2. 通信流机密性,
 3. 地址列表的正确性,
 4. 路由控制,
 5. 发布关于被多个用户偶尔连续重用的 PC 机,
 6. 消息和他们所要访问的的内容的自动确认,
 7. 消息复制检测, 重放阻止, 或其它的面向流的服务。

消息的发送者将决定是否对特定的消息进行保密增强服务。因此, 一个发送者必须能决定是否接收者具有处理保密增强邮件的能力。在一个一般的结构中, 这些机制将基于服务器的查询; 因此, 查询功能能被集成到一个 UA 避免增加电子邮件使用者的负担或不便。

4. 消息处理

4.1 消息处理总览

这个节提供了包括在电子邮件保密增强处理中一个组件和处理步骤的高级的总括和处理步骤。下面的小节将更详细定义这些过程。

4.1.1 密钥类型

一个两级的密钥层次被用于支持保密增强消息的传送:

1. 数据加密密钥 (DEKs) 被用于消息文本加密和 (在一组算法中可以进行一定的选择) 用于消息完整性检查的计算。DEK 为每个被传送的消息个别的产生; 数据加密密钥的预分发不需要支持保密增强的消息的传送。
2. 交换密钥 (IKs) 被用于加密在消息中传送的数据加密密钥。一般地, 同样的 IK 将被用于所有的从一个给定的发送者到一个给定接收者的所有的消息。每个被发送的消息包括一个被用于消息加密和 / 或 MIC 计算的数据加密密钥的表示, 这个密钥被每个命名接收者的个别交互密钥加密。与 “X-Sender-ID:” 和 “X-Recipient-ID:” 域相关的表示, 允许每个个别接收者标识被用于加密接受者使用 DEKs 和 / 或 MIC 的 IK。给一个适合的 IK, 一个接收者能解密相应的被传送的 DEK 表述, 产生用于消息文本解密和 / 或 MIC 验证的 DEK。一个 IK 定义的不同取决于用于 DEK 加密的是对称或非对称密码系统。
 - 2a. 当使用对称密码, 一个 IK 是一个发送者和接收者共享单独的对称密钥。在这种情况下, 相同的 IK 被用于加密传送的 DEK 和 MICs。版本 / 生命周期信息和与发送者和接收者有关的 IA 验证信息必须被串联为了充分的验

证一个对称 IK。

- 2b. 当使用非对称密码, 用于加密 DEK 的是接收者 IK 的公共组件。被用于加密 MIC 的是发送者 IK 的私有组件。因此每个消息只需要包括一个被加密的 MIC, 而不是每个接收者一个。这些 IK 中的每一个能被充分标识在 “X-Recipient-ID:” 或 “X-Sender-ID:” 域。

4.1.2 处理过程

当保密增强处理操作在一个发出的消息上, 一个 DEK 被产生由于消息加密和 (如果被选择的 MIC 算法需要一个密钥) 各种 DEK 被产生用于 MIC 计算。当一个消息的所有内容不需要加密时无需产生 DEK, 如果一个被选的 MIC 计算不需要一个密钥。

一个 “X-Sender-ID:” 域被包括在提供用于消息处理的 IK 的一个验证组件的头中。IK 组件被每个个别命名的接收者选择; 一个相应的 “X-Recipient-ID:” 域, 在前面解释的 “X-Sender-ID:” 域用于标识每个 IK。每个 “X-Recipient-ID:” 域接着一个 “X-Key-Info:” 域, 传送一个被对应特定接收者 IK 加密的 DEK。当一个特定的接收者使用对称密钥管理, “X-Key-Info:” 域也传送被接收者的 IK 加密消息的 MIC。当使用非对称密钥管理, 一个优先的 “X-MIC-Info:” 域存储由发送者的私有组件加密的消息的 MIC。

采用了四个阶段的处理过程用于以一种通用的传送形式表示被加密的消息文本和使被加密在一个主机计算机类型的消息被解密在不同的计算机类型。以本地形式接收的明文消息, 使用主机本地的字符集和行表示。本地形式被转换为一个规范的消息文本表示, 与内部的 SMTP 的消息文本的表示形式相同。这个规范的表示形成了 MIC 计算和加密处理的输入。

为了加密, 规范的表示按照加密算法的要求填充。被填充的规范的表示被加密 (除了那些明确表示无需加密的域)。被加密的文本 (以及无需加密的域的规范的表示) 被编码成一个可打印的形式。可打印的形式由一个严格的字符集组成, 这个字符集是每个站点通用的, 不会被在 MTS 实体内和之间的处理破坏。

编码过程的输出结合了带有密码控制信息的头域集。被传送给电子邮件系统的结果被封装为被传送文本部分。

当一个保密增强消息被处理, 文本中的密码控制域提供给被鉴别的接收方需要的信息。首先, 可打印的编码被转换为一个位串。被传送的消息的加密的部分被解密。MIC 被鉴别。规范的形式被转换为接收者本地的形式, 不需要和发送者的本地形式相同。

4.2 加密算法和模式

针对本文档的目的以 ANSI X3.92-1981 形式定义的块密码算法 DEA-1 将被用于消息文本的加密。DEA-1 与数据加密标准相同 (DES), 被定义在 FIPS PUB 46[3]。当被用于文本加密, DEA-1 将被用于密码块链接模式 (CBC), 定义在 ISO IS 8372[4]。标识字符串为 “DES-CBC”, 被定义在 RFC-1115, 表示这个算法/模式组合。CBC 模式被定义在 IS 8372 与在 FIPS PUB 81[5]和 ANSI X3.106-1993[16]中的定义相同。其他算法的使用和/或消息文本处理的模式将要求逐个的研究决定应用性和限制。此外在本文档中赞同使用的算法和模式将在后续文档到 RFC-1115 中定义。

为每个保密增强电子邮件消息产生新的伪随机初始向量是发送者的责任除非整个消息不需要加密。参考资料[17]的 4.3.1 节提到这一要求,甚至在个别 DEKs 被产生用于个别消息的情况下, IV 也将随着消息被传送。

一定的操作要求一个被传送的密钥被一个交互的密钥加密。头部内容指出了 IK 被用于加密的模式。RFC1115 规定了加密算法/模式的标识,包括 DES-ECB, DES-EDE, 和 RSA。所有使用对称密钥管理的实现应该支持 DES-ECB IK 的使用,所有使用非对称密钥管理的实现应该支持 RSA IK 的使用。

RFC-1114, 当前和这个文档一起发放,规定了非对称的基于证书的密钥管理过程支持定义在这个文档中的消息处理过程。消息处理过程也能和对称密钥管理一起使用,合适的对称 IKs 通过带外通道预先发放。强烈推荐支持在 RFC1114 中定义的非对称方法。

4.3 保密增强消息转换

4.3.1 约束

一个电子邮件加密机制必须与底层电子邮件功能的透明约束相兼容。这些约束一般被建立在预期的用户要求和预期的端点和传输设备的特色,加密机制必须也与所交互的计算机系统的本地规范相兼容。在我们的建议中,一个规范的步骤是抽象我们的本地规范和操作一个后续的编码步骤以与底层邮件传输媒体 (SMTP) 特色相一致。编码与 SMTP 的约束相一致,用于支持人与人之间的通讯。SMTP 的规则也以规范的处理过程独立使用。RFC-821's 的 4.5 节详细说明了 SMTP's 的透明约束。

准备一个进行 SMTP 传送的消息必须满足以下的要求:

1. 所有的字符必须是 7 位 ASCII 码中的一员。
2. 文本行,通过字符对<CR><LF>划界,不能超过 1000 个字符长。
3. 因此字符串<CR><LF>, <CR><LF>指出了消息的结束,它必须在文本中先于消息的结束出现。

尽管 SMTP 规定了一个标准的行分界符的表述,大量的系统使用一个不同的本地的行分界的表述。例如,在邮件中使用的行分界符<CR><LF>进入 UNIX 操作系统被转换为单一的<LF>s 作为邮件的分界符被写到本地的邮件箱文件中。邮件中的行引入到面向记录的系统(例如 VAX VMS)可以被目的 SMTP 服务器转换成合适的记录。因此如果加密处理产生<CR>s 或 <LF>s,这些字符可能被一个使用不同行分界规则的接收用户代理进程获取。也可能 Tabs 和空格的转换可以在 SMTP 内部和本地格式映射的过程中被处理;这是一个本地选择的问题。如果这样的转换改变了被传送的密文的形式,解密将不能产生被传送的明文,被传送的 MIC 将不能和在目的计算机上计算出来的 MIC 相比较。

在采用 EBCDIC 作为本地字符集的系统被 SMTP 服务器操作的转换甚至有更严重的影响,因此从 EBCDIC 到 ACSII 的转换是一个信息损失的转换。在原则上,在 SMTP 内部规范 ASCII 消息表示和本地格式之间的映射转换功能可以从 SMTP 服务器上移到用户代理上,给定的管理 SMTP 服务器的手段不应该再进行那种转换。这个方法有很大的缺陷:内部的文件(例如,邮件箱)格式和它所驻留的本地文件形式不兼容。此外,它将要求修改 SMTP 服务器,邮件将以一种与现在被传递的不同的表述被传递给 SMTP 服务

器。

4.3.2 建议

我们建议支持保密增强邮件通过一个中间转换可以发生的环境, 编码邮件以一种统一的表述风格通过保密增强用户代理不论他们的系统采用何种本地字符集。这种编码的形式被用于表示从发送者到接受者的邮件文本, 但是编码未被用于封装邮件传输头或去封装插入到载有保密增强用户代理之间的控制信息的头中。编码的特色使在预期的发送者和接收者用户代理之间的转换不会阻止一个被编码的消息在它的目的地被正常解密。

一个发送者从加密处理可以排除一个消息一个或更多的部分, 但是鉴别处理总是被应用到这个消息文本。将一定比例的消息排除在加密处理之外是被明确要求的; 默认的加密被应用到整个消息文本。规定这一排除操作的用户级的分界符是本地的操作, 因此可以在发送者和接受者之间变动, 但是所有的系统应该提供一个手段明确的标识被排除在加密处理之外的域。

外部的保密增强消息进行四步转换, 在下面的四个小节描述。

4.3.2.1 步骤一: 本地形式

消息文本被以本地的系统的字符集创建, 行的分界与本地规则相一致。

4.3.2.2 步骤二: 规范形式

整个文本消息, 包括需要加密处理的部分和不需要加密处理的部分, 被转换为一个通用的规范形式, 与在 RFC-821 和 RFC-822[10]中定义的 SMTP 之间的表述类似 (ASCII 码字符集, <CR><LF>行分界符)。这个处理要求在本地字符集是 ASCII 码时进行的转换最小。(注意: 规范编码处理的输出将永远不会被直接传给 SMTP, 而是传给保密增强编码处理的后续步骤, 圆点填充的转换在 RFC-821 中讨论, 因此一个消息在加密前被转换成一个标准的字符集和表述, 它能被解密并且它的 MIC 能在任何类型的目的主机计算机上被验证。解密和 MIC 验证在转换 (将消息转换到一个规定的本地形式) 前进行。

4.3.2.3 步骤三: 鉴别和加密

规范形式被输入到被选择的 MIC 计算算法中为了计算消息的一组完整性检查值。在提交给 MIC 计算算法之前没有值被填充到规范形式, 尽管一定的 MIC 算法将在计算 MIC 的过程中将进行他们自己的填充。

应用到规范形式的填充在 DEA-1 CBC 模式的加密中是需要的, 如下: 加密字节数由从总长中减去无须加密的字节长度来决定。在加密的文本需要时 16 进制的字符 FF 与被填充的字符一起被附加到规范形式中, 用 8 字节加密量的整数值来添满。如果所加密的字符数已经是 8 的整数倍则无须填充。16 进制 FF (一个值超出 7 位 ASCII 字符集) 填充

值允许在没有包括一个明确的填充个数的指示时与有效的数据相区别。

没有被排除在加密之外的消息域被加密。为了支持可选的加密处理，一个实现必须保留加密区域和非加密域的内部的标识，以便这些域在步骤四定义的编码过程中能被适当的分界。如果无须加密的域插入到加密域之间，密码状态（例如，IVs 和进行加密的字符）被保留在这个被排除的域之后继续。

4.3.2.4 步骤四：可打印的编码

继续从左到右，步骤三的位串被编码成在所有的站点都通用的字符表示，尽管不必是同样的位模式（例如，尽管字符“E”在基于 ASCII 码字符的系统被表示为 16 进制的 45 在基于 EBCDIC 的系统被表示为 16 进制的 C5，两种表述的本地意义是相同的）。这个编码步骤在所有的保密增强消息中进行，即使整个消息不需要加密。

一个国际的字符 IA5 的一个被使用 64 字符集，每个可打印的字符由 6 位表示。（建议的字符集在 IA5 和 ASCII 中的表示是相同的。）两个额外的字符，“=”和“*”被用于表示特殊的处理功能。字符“=”被用于在可打印的编码过程中填充。字符“*”被用于分界无须加密域的开始和结束。编码功能的输出被划分为文本行（使用本地规范），除了最后一行每一行确切地包含 64 个可打印的字符，最后一行包含 64 或少于 64 的可打印字符。（这一行长度是易被打印的并保证满足 SMTP 的每行最多 1000 个字符的限制。）

编码过程将输入的每组 24 位表示为输出的 4 个字符。从左到右的一个 24 位输入组从步骤三的输出中获得，每 6 位组被用作一个 64 位可打印的字符的索引。被索引指定的字符被放置在输出串中。这些字符，被标识在表 0 中，被选择作为通用的表示，对 SMTP 有特殊意义的字符被排除（例如，“.”，“<CR>”，“<LF>”）。

如果输入的一组少于 24 位进行特殊的处理，要么在一个消息尾（当可选的加密功能被调用）要么在一个被加密的域或未加密的域的末端。全部的编码总是在消息末和分界符“*”被输出初始或终止一个无需加密的块之前被完成。当输入的一个组消息少于 24 位，位 0 被填充使消息是 6 的整数倍。不需要表示实际输入数据的输出字符位被置为字符“=”。因此所有的通用编码的输出是一个 8 位字节的整数，只有下面的情况能出现：（1）编码输入的最后量是 24 位的整数倍；这，编码输出的最后的单元是不含“=”填充的 4 的整数倍，（2）编码输入的最后量确定为 8 位；这，编码输出的最后的单元将两个字符并跟有两个“=”的填充字符，或（3）编码输入的最后量确定为 16 位；这，编码输出的最后的单元将是三个字符并跟有一个“=”填充字符。

4.3.2.5 转换概述

总的说，发送的消息服从下面的转换式：

$$\text{Transmit_Form} = \text{Encode} (\text{Encipher} (\text{Canonicalize} (\text{Local_Form})))$$

相反的操作以相反的顺序转换处理接收的保密增强邮件：

$$\text{Local_Form} = \text{Decanonicalize} (\text{Decipher} (\text{Decode} (\text{Transmit_Form})))$$

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0

2 C	19 T	36 k	53 l
3 D	20 U	37 l	54 2
4 E	21 V	38 m	55 3
5 F	22 W	39 n	56 4
6 G	23 X	40 o	57 5
7 H	24 Y	41 p	58 6
8 I	25 Z	42 q	59 7
9 J	26 a	43 r	60 8
10 K	27 b	44 s	61 9
11 L	28 c	45 t	62 +
12 M	29 d	46 u	63 /
13 N	30 e	47 v	
14 O	31 f	48 w	(pad) =
15 P	32 g	49 x	
16 Q	33 h	50 y	(1) *

(1) The character "*" is used to enclose portions of an encoded message to which encryption processing has not been applied.

Printable Encoding Characters

Table 1

注意本地的形式和转换消息到标准形式及从标准形式转换的功能可以在发送者和接收者之间无信息损失的变动。

4.4 封装机制

在一个由电子邮件传输系统解释的头的一个封装层里的保密增强消息的封装比一个简单的被加密或携带密码控制信息的头内的一定域的简单的处理有更多的优势。封装提供了一般性和那些在传输时被转换的消息的用户对用户级的隔离的域。被插入到加密/鉴别过程中的所有的域被放置在封装的头中。这个功能为只接收文本没有从输入文件或从其它程序获得头域的邮件处理程序提供了兼容性。此外，保密增强处理能被递归使用。关于MTS，合并到密码鉴别或加密处理的信息将驻留到一个消息的文本部分，而不是头部分。

用于保密增强邮件的封装机制来自于RFC-934[11]中的叙述，这一叙述是基于internet社区消息摘要处理的先例。准备的用于加密或鉴别的一个用户消息将被转换为图1中的表述。

作为一般的设计原则，敏感的数据通过将数据导入到被封装的文本而不是通过将数据导入到封装的头里的域的方法来进行保护。潜在的敏感头信息的域可以包括这些域例如：“Subject:”，包含的内容对于端到端的用户和内部的用户有意义。应用保护的域集（可能是空的）由用户选择。不过强烈推荐所有的信息应当在封装文本里复制“X-Sender-ID:”和“X-Recipient-ID:”域的拷贝。

如果一个用户希望对头域公开保护, 他们必须只出现在被封装的文本中而不在被封装的头中。如果公开保护要求一个消息的主题标识, 推荐封装的头包含一个“Subject:”域指出“被加密的邮件跟随”。

如果要求一个头信息的被鉴别的版本, 除了本身在封装头里的数据, 包含在被封装的本文里的数据也能被复制。例如, 一个发送者希望提供接收者一个在一系列被封装的文本中的包括一个时间戳或消息计数域的被保护的消息的位置的标识。

一个与带有消息封装机制功能的保密增强邮件的完整性有关的特殊的点是值得注意。被选择用于传输编码的 IA5 子集有目的的排除了字符“-”, 因此被封装的文本可以明确

的同一个消息的结束封装的边界相区分 (Post-EB) 不用求助于字符的填充。

Enclosing Header Portion

(Contains header fields per RFC-822)

Blank Line

(Separates Enclosing Header from Encapsulated Message)

Encapsulated Message

Pre-Encapsulation Boundary (Pre-EB)

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Encapsulated Header Portion

(Contains encryption control fields inserted in plaintext.
Examples include “X-DEK-Info:”, “X-Sender-ID:”, and
“X-Key-Info:”.

Note that, although these control fields have line-oriented representations similar to RFC-822 header fields, the set of fields valid in this context is disjoint from those used in RFC-822 processing.)

Blank Line

(Separates Encapsulated Header from subsequent encoded Encapsulated Text Portion)

Encapsulated Text Portion

(Contains message data encoded as specified in Section 4.3; may incorporate protected copies of enclosing and encapsulated header fields such as “Subject:”, etc.)

Post-Encapsulation Boundary (Post-EB)

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Message Encapsulation

Figure 1

4.5 邮件列表的邮件

当邮件被定位到邮件列表,可以采用两种不同的方法: **IK-per-list** 方法和 **IK-per-recipient** 方法。选择依赖于对于发送者可用的信息和发送者的兴趣。

如果一个消息的发送者定位一个消息到一个列表名或别名,表示他将一个 **IK** 和那个名字或别名作为一个实体结合使用 (**IK-per-list**),而不是决定到它的目的地的名字或别名。因此 **IK** 必须对列表的所有成员都是可用的。对于非对称密钥管理的情况,列表的私有组件必须的列表的所有成员都是可用的。另一种是凭借远端爆增站点发送消息的一般的情况,此时到这样的列表的发送者可以不认识个别的接收者。不幸的是,它暴露了共享的 **IK**,使它的更新变得困难。此外, **IK-per-list** 方法的使用允许列表的 **IK** 的任何所有者可以伪装成其他的为了鉴别的目的到这个列表发送者。

与此相对,如果一个消息的发送者可以将目的邮件列表扩展为自己的组成部分并选择这么做 (**IK-per-recipient**),消息的 **DEK** (和,在对称密钥管理的情况下, **MIC**) 将用每个接收者的 **IK** 加密并且这些被加密的表述将被合并到被传送的消息中。注意每个接收者的加密只要求在“**X-Key-Info:**”中携带相对小的 **DEK** 和 **MIC**,不像加密消息文本时一般要求更大。尽管更多的 **IK** 在 **IK-per-recipient** 方法中被处理,一对 **IK** 能被个别调用而且只拥有一个 **IK** 并不能使另一个列表上的使用者成功的进行伪装。

4.6 被封装的头域小结

这部分总结了在保密增强处理的过程中被添加到消息中的被封装的头域的语法和语义。这些头域分三组出现。正常的,一组将按顺序出现在被封装的头域,尽管在每一组里不是所有的域都会出现在所有的消息中。在某些特定情况下,这些域被复制到被封装的消息文本中也被包括到被封装的头中。图 2 和 3 是被封装的消息的小例子。图 2 假设使用对称的密钥管理。图 3 假设说明了使用非对称的密钥管理的被封装的消息的示例。

如果没有其他的特殊的规定,所有的域参数以一种区分大小写的风格处理。在大多数情况下,数字量以连续的十六进制数出现在头域中,每个数字通过从“0”到“9”或大写的“A”到“F”的字符来表示。因此公钥证书和使用非对称算法加密的量尺寸比较大,一个更节省空间的编码技术的使用对这样的量是合适的,而且定义在本文档 4.3.2.4 节的用可打印的字符表示 6 位的编码机制被采用。在图 3 中显示的例子显示的被加密的非对称量(例如,“**X-Mic-Info:**”,“**X-Key-Info:**”)用 64 个可打印的字符表示,对应 384 位。带有非对称加密量的域也说明了定义在 RFC822 中的 3.1.1 节的折叠的使用。

```
-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----  
X-Proc-Type: 3,ENCRYPTED  
X-DEK-Info: DES-CBC,F8143EDE5960C597  
X-Sender-ID: linn@ccy.bbn.com::  
X-Recipient-ID: linn@ccy.bbn.com:ptf-kmc:3
```

X-Key-Info: DES-ECB,RSA-MD2,9FD3AAD2F2691B9A,B70665BB9BF7CBCD,
A60195DB94F727D3

X-Recipient-ID: privacy-tf@venera.isi.edu:ptf-kmc:4

X-Key-Info: DES-ECB,RSA-MD2,161A3F75DC82EF26,E2EF532C65CBCFF7,
9F83A2658132DB47

LLrHB0eJzyhP+/fSStdW8okeEnv47jxe7SJ/iN72ohNcUk2jHEUSoH1nvNSIWL9M
8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHUIBLpvXR0UrUzYbkNpk0agV2IzUpk
J6UiRRGcDSvzrsoK+oNvqu6z7Xs5Xfz5rDqUcMIK1Z6720dcBWGGsDLpTpSCnpot
dXd/H5LMDWnonNvPCwQUHt==

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Example Encapsulated Message (Symmetric Case)

Figure 2

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

X-Proc-Type: 3,ENCRYPTED

X-DEK-Info: DES-CBC,F8143EDE5960C597

X-Sender-ID: linn@ccy.bbn.com::

X-Certificate:

jHUIBLpvXR0UrUzYbkNpk0agV2IzUpk8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIk
YbkNpk0agV2IzUpk8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHUIBLpvXR0UrUz
agV2IzUpk8tEjmFjHUIBLpvXR0UrUz/zxB+bATMtPjCUWbz8Lr9wloXIkYbkNpk0

X-Issuer-Certificate:

TMtPjCUWbz8Lr9wloXIkYbkNpk0agV2IzUpk8tEjmFjHUIBLpvXR0UrUz/zxB+bA
IkjHUIBLpvXR0UrUzYbkNpk0agV2IzUpk8tEjmF/zxB+bATMtPjCUWbz8Lr9wloX
vXR0UrUzYbkNpk0agV2IzUpk8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHUIBLp

X-MIC-Info: RSA-MD2,RSA,

5rDqUcMIK1Z6720dcBWGGsDLpTpSCnpotJ6UiRRGcDSvzrsoK+oNvqu6z7Xs5Xfz

X-Recipient-ID: linn@ccy.bbn.com:RSADSI:3

X-Key-Info: RSA,

IBLpvXR0UrUzYbkNpk0agV2IzUpk8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHU

X-Recipient-ID: privacy-tf@venera.isi.edu:RSADSI:4

X-Key-Info: RSA,

NcUk2jHEUSoH1nvNSIWL9MLLrHB0eJzyhP+/fSStdW8okeEnv47jxe7SJ/iN72oh
LLrHB0eJzyhP+/fSStdW8okeEnv47jxe7SJ/iN72ohNcUk2jHEUSoH1nvNSIWL9M
8tEjmF/zxB+bATMtPjCUWbz8Lr9wloXIkjHUIBLpvXR0UrUzYbkNpk0agV2IzUpk
J6UiRRGcDSvzrsoK+oNvqu6z7Xs5Xfz5rDqUcMIK1Z6720dcBWGGsDLpTpSCnpot
dXd/H5LMDWnonNvPCwQUHt==

-----PRIVACY-ENHANCED MESSAGE BOUNDARY-----

Example Encapsulated Message (Asymmetric Case)

Figure 3

尽管被封装的头域类似 RFC-822 的头域，他们之间并没有交集而且不是使用同样的处理密封头域的分析器。对被封装头域字典分析的复杂度明显比 RFC-822 头域的分析复杂度要小。例如，许多对于在依据造句法的级别的 RFC-822 有特殊意义的字符在被封装的头域中没有这种特殊的意义。

当被封装的头域的长度比一行可打印的字符长度要长时，在 RFC-822 3.1.1 节中可以用空格折叠这个域。任何这样空格的加入不会被解释为这一子集的一部分。作为一个特殊的例子，由于公钥证书和使用非对称算法加密的量的长度，这些量经常需要被折叠成几行。为了以统一的方式使用这种折叠，这样一个量的位表示按顺序（最左边的位置先出现）被划分 0 或多于 384 位的组（对应 64 位可打印的字符的表示），最后一组可以是小于 384 位的任意的长度。

4.6.1 每个消息被封装的头域

这组被封装的头域包含在一个保密增强消息中出现不止一次的域，一般先于所有其他的被封装的头域。

4.6.1.1 X-Proc-Type 域

“X-Proc-Type:”被封装的头域，是所有的保密增强消息都必须具有的，标识对被传送的消息进行的处理类型。在一个消息中该域只出现一次；“X-Proc-Type:”域必须在第一个出现在被封装的消息头中。

“X-Proc-Type:”域必须有两个子域，被一个逗号分隔。第一个子域是一个十进制数用于区别不兼容的被封装的头域说明可能会在以后对这个标准进行修改后出现。按照本文档消息处理将带有子域值 3 用于与以前的 RFCs 989 和 1040 相区别。

第二个子域可以假设为两个字符串值中的一个：“ENCRYPTED”或“MIC-ONLY”。如果一个消息的被封装的文本需要加密，“X-Proc-Type:”域的第二个子域必须规定“ENCRYPTED”。“MIC-ONLY”的规定，在和密钥管理和 MIC 算法选择相关联时，允许一些未进行加密的域从被封装的头域中忽略。尤其“X-Recipient-ID:”和“X-Key-Info:”在非对称密钥算法被使用时能被接收者忽略。假设当前使用一个无密钥 MIC 计算算法，“X-DEK-Info:”域可以被所有的“MIC-ONLY”消息忽略。

4.6.1.2 X-DEK-Info 域

“X-DEK-Info:”被封装头域标识消息文本加密算法和模式，也携带用于消息加密的初始向量。在消息里“X-DEK-Info:”域不止出现一次，在“X-Proc-Type:”域里规定了“MIC-ONLY”的消息不包含此域。

“X-DEK-Info:”域带有两个参数，用逗号分隔。对于本 RFC 的目的，第一个参数必须是字符串“DES-CBC”，表示使用 CBC 模式的 DES 算法。第二个参数代表了一个 64 位的初始向量（IV）以连续的 16 进制的形式表示。后续的 RFC1115 修订版将规定可能作为这

个域的第一个参数出现的任何额外的值。

4.6.2 一般每个消息被封装的头域

这个被封装的头域组包含在每个消息中出现不止一次的头域。依赖使用的密钥管理选项，这些域中的一些可以在某些消息中不出现。在一个消息中“X-Sender-ID”域可以出现不止一次如果对于不同的接收者必须使用不同的面向发送者的 IK 组件（也许对应于不同的版本）。在这种情况下后来的出现取代以前的出现。如果在一个简单消息里使用对称和非对称的密钥分发的混合。对于密钥分发技术的每个接收者应该被组合在一起简化分析。

4.6.2.1 X-Sender-ID 域

所有的保密增强消息都要求有“X-Sender-ID:”被封装的头域，标识一个消息的发送者并提供发送者的 IK 标识组件。它应该在被封装的文本内被复制。IK 标识组件被包含在“X-Sender-ID:”域与所有后续的“X-Recipient-ID:”相关联直到另一个“X-Sender-ID:”域出现；一般的情况是只有一个“X-Sender-ID:”域在任意“X-Recipient-ID:”域前出现。

“X-Sender-ID:”域包含一个实体标识子域，一个（可选的）发行机构子域，和一个（可选的）版本/满期子域。这些可选的域可以被忽略如果他们的出现对于后续的“X-Recipient-ID:”域所带有的信息来说是多余的。这在使用对称密码作为密钥管理的情况下是经常的情况。这些子域通过冒号分隔，也可以带有空格。

在 5.2 节，交互密钥，讨论了这些子域的语义并规定了他们选择的字符的形式。注意多个“X-Sender-ID:”域可以出现在一个简单的被封装的头中。所有的“X-Recipient-ID:”域在大多数情况下接着“X-Sender-ID:”域；“X-Recipient-ID:”域出现在“X-Sender-ID:”域之前是不合法的。

4.6.2.2 X-Certificate 域

“X-Certificate:”域只在非对称密钥管理被用于一个或多个消息的接受者时被使用。为了便于接收者的处理（至少超过一个一般的路径服务器可用性），强烈推荐在所有的消息中包括这个域。这个域以数量的形式传送了发送者的证书，以定义在 4.3.2.4 节中的编码的形式表示。一个证书的语法定义在 RFC-1114 中。在“X-Certificate:”域中携带的证书和“X-Sender-ID:”域和“X-Recipient-ID:”域一起在非对称密钥管理被使用时使用。

4.6.2.3 X-MIC-Info 域

“X-MIC-Info:”只在至少一个消息的接收者使用非对称密钥管理时才使用，带有三个参数，通过逗号分隔。第一个消息标识在 MIC 被计算时使用的算法；RFC-1115 规定了可接受的 MIC 算法标识集。第二个参标识 MIC 被加密时使用的算法；在本文档中必须出现在

RFC-1115 中描述的字符串“RSA”，标识 RSA 算法。第三个参数是一个 MIC。

非对称的加密使用发送者的私钥。正如本文档前面所述，非对称被加密的 MIC 使用描述在 4.3.2.4 节中的技术来表示。

“X-MIC-Info:”域将立即出现在“X-Sender-ID:”域和“X-Certificate:”域或“X-Issuer-Certificate:”之后。类似“X-Sender-ID:”域，一个“X-MIC-Info:”域由所有的使用非对称密钥的接收者使用。

4.6.3 不定出现的头域

这组被封装的头域包含在消息中出现次数不定的域，出现的次数从 0 到非零的值变动与接收者的数目无关。

4.6.3.1 X-Issuer-Certificate 域

“X-Issuer-Certificate:”被封装的域只在至少一个消息的接收者使用非对称密钥管理时是有意义的。一个典型“X-Issuer-Certificate:”域包含拥有公钥组件的证书，这个证书被用于签包含在“X-Certificate:”域中的证书，接收者通过证书的认证路径链使用。其他的典型的代表认证链上高一级的证书的“X-Issuer-Certificate:”域，也可以被一个发送者包括。被包括的“X-Issuer-Certificate:”域的顺序不需要对应认证路径的顺序；路径的顺序一般可以与不同的接收者的观点不同。关于认证路径的更多的信息可以在 RFC-1114 中发现。

证书以定义在“X-Certificate:”域中相同的方式表示，任何“X-Issuer-Certificate:”域一般将直接跟随“X-Certificate:”域。“X-Issuer-Certificate:”域这个域甚至在使用非对称密钥管理时也是可选的，尽管在使接收者能获取发行者的证书的的二选一的路径服务器缺乏时强烈推荐使用这个域。

4.6.4 每个接收者被封装的头域

这组被封装的头域对于每一个消息的命名接收者来说一般出现一次。在特殊的情况下，这些域在发送给使用非对称密钥管理的接收者一个“MIC-ONLY”消息的情况下可以被忽略，给定的被选择的 MIC 算法是无密钥的。

4.6.4.1 X-Recipient-ID 域

这个“X-Recipient-ID:”标识了一个接收者并提供了接收者 IK 标识组件。一个“X-Recipient-ID:”被每个命名的接收者包括。它应该被复制在被封装的文本中。这个域包含一个实体标识子域，一个发行机构子域，和一个版本子域。子域通过冒号分界，可以跟空格。

5.2 节，交互密钥，讨论了子域的语义并规定了他们被选择的字符。所有“X-Recipient-ID:”域跟在最接近的“X-Sender-ID:”域之后；“X-Recipient-ID:”域出现

在“X-Sender-ID:”域之前是不合法的。

4.6.4.2 X-Key-Info 域

一个“X-Key-Info:”被包含在每一个消息的命名接收者中。每一个“X-Key-Info:”域跟在最近的“X-Recipient-ID:”域之后;通常,一个“X-Key-Info:”域将立即跟着它的相关的“X-Recipient-ID:”域。对于一个特殊的接收者这个域的参数对于对称和非对称的密钥管理是不同。

4.6.4.2.1 对称密钥管理

当对一个给定的接收者使用对称密钥管理,“X-Key-Info:”被封装的头域传送 4 个条目,通过逗号分隔:一个 IK 使用标识,一个 MIC 算法标识,一个 DEK 和一个 MIC。IK 使用标识符标识了算法和被标识的 IK 用于一个特殊的接收者的 DEK 加密的模式。对于对称密钥管理被使用的接收者,它可以假设保留的字符串值为“DES-ECB”或“DES-EDE”,定义在 RFC-1115 中。

MIC 算法标识符标识用于特殊接收者的 MIC 计算算法;这个子域的值被定义在 RFC-1115 中。DEK 和 MIC 使用前面被“X-Sender-ID:”和“X-Recipient-ID:”域标识的 IK 来加密;他们以两个连续的 16 进制字符串来表示,通过一个逗号分隔。

当 DEA-1 被用于消息文本的加密,DEK 将是 16 个 16 进制数字。(对应一个 64 位的密钥);这个子域能被扩展为 32 位 16 进制数字(对应一个 128 位密钥)如果需要支持其他的算法。

MIC 的对称加密也以和消息 DEK 的加密的相同模式来加密。被加密的 MICs,像被加密的 DEKs,以连续的 16 进制字符串来表示。MIC 的大小依赖于规定在 MIC 算法标识子域的 MIC 算法的选择。

4.6.4.2.2 非对称密钥管理

当对一个给定的接收者使用非对称密钥管理,“X-Key-Info:”域传送两个量,通过逗号分隔。第一个参数是一个 IK 使用标识符标识加密 DEK 的算法(和模式,如果可用);本文档,IK 使用标识符子域总假设保留字符串为“RSA”(定义在 RFC-1115)对于使用非对称密钥管理的接收者,表示 RSA 算法的使用。第二个参数是一个 DEK,在接收者的公共组件下加密(使用非对称加密)。

在本文档中我们采用术语“私有组件”和“公共组件”参考对称密码系统中分别保持保密和使公共可用的两个量。这个规定被采用避免因为术语“密钥”用于指代私有组件和对密码中的密钥引起的困惑。

正如在本文档前面所讨论的,非对称被加密的 DEK 使用在 4.3.2.4 节所描述的方法表示。

5. 密钥管理

几个密码元素被使用支持保密增强消息处理的过程。假定了一组基本的元素。数据加密密钥(DEKs)被用于加密消息文本和(用于一些 MIC 计算算法)在消息完整性检查(MIC)计算过程。交互密钥(Iks)被用于加密和消息一起传送的 DEKs 和 MICs。在一个基于证书

的非对称密钥管理的结构中,证书被用于作为一个提供实体的公共组件和被中央权威机构安全绑定的信息的手段。在这一节的剩余部分提供了关于这些结构的信息。

5.1 数据加密密钥 (DEKs)

数据加密密钥 (DEKs) 被用于加密消息文本和 (带有一些 MIC 计算算法) 消息完整性检查的计算。强烈推荐 DEKs 对每个消息被产生使用一次。一个被传送的消息将合并一个被对每个命名接收者的合适的交互密钥加密的 DEK。

DEK 产生可以要么通过密钥分发中心 (KDCs) 或通过端系统。专门的 KDC 系统可以实现比端系统支持的算法强的随机 DEK 产生算法。另一方面,分散允许端是相对独立的,减少了必须放在除了消息的接收者和发送者的组件的信任级别。此外,在端点的分散的 DEK 的产生减少了发送者为了发送邮件进行实时服务器查询 (潜在的唯一的) 的频率,加强交互可用性。

当对称密码被使用时,一个基于 KDC 集中产生的优势是 DEKs 能在被消息的接收者的 Iks 加密后被返回给端点而不是提供 IKs 给发送者。这减少了 IK 的暴露并简化了端点密钥管理的要求。如果使用非对称密钥管理这个方法没什么价值,因此每个接收者公共 IK 组件被认为一般是可用的而且每个发送者的私有 IK 组件不需要和 KDC 共享。

5.2 交互密钥 (Iks)

交互密钥 (IK) 组件被用于加密 DEKs 和 MICs。一般,IK 间隔尺寸是除了发送给包含多个用户的地址列表的邮件的每个对等用户级别。对于使用标准的密码进行保密增强电子邮件交互有两个主要的原则,首先必须处理公共的 IK 组件 (当使用对称密钥管理) 或补充的 IK 组件 (当使用非对称密钥管理)。当使用对称密码,IK 由一个单一的组件构成,被用于加密 DEKs 和 MICs。当使用非对称密码,一个接收者的公共组件用做一个 IK 加密 DEKs (一个相反的转变只由接收者处理对应私有组件),发送者的私有组件被用于加密 MICs (一个相反的转变由所有的接收者操作,因此发送者的证书提供了发送者的必要的公共组件)。

而本文档没有规定交互密钥被提供给合适的使用者的手段,注意这些可能被集中 (例如,通过密钥管理服务器) 或分散 (例如,通过对等协商和直接在用户中分发) 的手段是有用的。在任何情况下,任何给定的 IK 组件和一个对应的发行机构 (IA) 相关。当基于认证的在 RFC-1114 中讨论非对称密钥管理被采用,IA 功能通过一个证书机构实现 (CA)。

当一个 IA 产生和分发一个 IK 组件,相关的控制信息被提供指导如何使用 IK。为了选择用于消息加密的合适的 Iks,一个发送者必须保留一个在 IK 组件和与之相关的接收者的通信。终止时间信息必须被保留,以便可以使存储的入口无效并被合适的替代。

因此一个消息可以被多个 IK 组件标识发送给相应的多个接收者,每个接收者的用户代理必须能够决定接收者需要的 IK 组件。此外,如果当一个消息到达时接收者的数据库里没有相应的 IK 组件,接收者必须能鉴别需要的 IK 组件并鉴别与 IA 相关的 IK 组件。注意不同的 IK 可以在一对通信者之间被用于不同的消息。考虑,例如,一个从 A 发送到 B 的消息和另一个从 A 发送到 B 所在的邮件列表的消息 (使用 IK-per-list 方法)。第一个消息将使用分别与 A 和 B 相关联的 IK 组件,但是第二个将使用在列表成员之间共享一个 IK 组件。

当一个保密增强消息被发送,一个用于加密 DEK 和 MIC 的 IK 指示必须被包括。到此

为止, “X-Sender-ID:” 和 “X-Recipient-ID:” 被封装的头域提供下面的数据:

1. 相关发行机构的鉴别 (IA 子域)
2. 与一个特殊 IK 组件相关的一个实体的鉴别 (实体标识符或实体标识子域)
3. 版本/满期子域

冒号被用于在一个 “X-Sender-ID:” 或 “X-Recipient-ID:” 中进行分界。IA, EI, 和版本/满期子域从一个严格的字符集中产生, 通过下面的 BNF 表述 (使用定义在 RFC-822, 第 2 节和 3.3 节中的符号):

```
IKsubfld      :=      1*ia-char

ia-char       :=      DIGIT / ALPHA / "" / "+" / "(" / ")" /
                      ";" / "." / "/" / "=" / "?" / "-" / "@" /
                      "%" / "!" / "'" / "_" / "<" / ">"
```

一个 “X-Recipient-ID:” 域的示例如下:

X-Recipient-ID: linn@ccy.bbn.com:ptf-kmc:2

这个例子表明 IA “ptf-kmc” 已经发行了一个 IK 组件用于发送给linn@ccy.bbn.com的消息, 并且 IA 提供了数字 2 作为一个对于那个 IK 组件的版本标识符。

5.2.1 子域的定义

下面的几节定义了 “X-Sender-ID:” 和 “X-Recipient-ID:” 域。

5.2.1.1 实体标识符子域

一个实体标识符被构成作为一个 Iksubfld。更严格地, 一个实体标识符子域假设了下面的形式:

```
<user>@<domain-qualified-host>
```

为了支持通用的交互性, 必须假设一个命名信息的通用的形式。对于传送到更广的网络的转换本地主机名的安装的情况, 强烈推荐主机名被呈现给使用的 Internet。

5.2.1.2 发行机构子域

一个 IA 标识符子域被构成一个 Iksubfld。IA 标识符必须以一种确保唯一的方式被分配。这可以在一个集中或分层的结构上使用。

5.2.1.3 版本/满期子域

一个版本/满期子域被构成一个 Iksubfld。版本/满期子域格式可以在不同的 IAs 中变动, 但是必须满足一定的功能约束。一个 IA 的版本/满期子域必须能足够为一个给定被鉴别的实体区别被 IA 发行的 IK 组件集。单调增加的数的使用足以区别被一个 IA 提供给一个实

体的 IK 组件; 一个时间戳的使用又允许一个被指定给一个 IK 组件的满期时间或日期。

5.2.2 IK 加密期发行

一个 IK 的加密期部分是在密钥间接管理和撤回响应之间的折中规定, 它将不需要在一个使用 IK 组件加密的消息接收前永久的删除一个 IK 组件, 这将使这个消息永久不可识别。将需要获取一个过期的 IK 组件, 例如, 处理被一个已经超过一个期限不活动的用户(或系统)接收的邮件。为了使非常旧的 IK 组件被删除, 一个需要被加密的本地长时间存储的消息的接收者应该通过使用本地维护的 IK 重加密转换被用于消息文本加密的 DEK, 而不是依赖于 IA 无限期的维护旧的 IK 组件。

6. 用户命名

6.1 当前的方法

为了正确的选择相应的密钥对电子邮件的使用者进行唯一的命名是一个重要的课题并已进行了仔细的研究。我们当前的把 IK 组件和用户名联系起来的结构以一种通用的方式表示为 ([user@domain-qualified-host](#)), 依赖于以下的属性:

1. 通用的形式必须通过一个 IA 说明的当这个 IA 分发 IK 组件并当它处理被接收的 IK 组件和 IK 组件标识符时用户的代理可以识别。如果一个 UA 或 IA 以本地的形式使用地址不同于通用的形式, 它必须能在通用形式和本地表示之间进行明确的映射。
2. 通用形式, 当被一个发送者的 UA 处理时, 必须和被用户规定的一个接收者地址的形式有一个可以辨认的通信。

在整个 Internet 上保证这些属性是困难的。例如, 一个对在一个组织内部使用的本地形式和被用于整个 Internet 邮件传输使用的通用形式之间进行转换的邮件传输系统可能违反属性 2

6.2 发行考虑

平面的(非层次)电子邮件使用者标识符的使用, 与用户所在的主机无关, 可以提供价值。当路径服务器变得更普遍, 寻找需要的基于这种属性的接收者对于可能成为的发送者来说是合适的。个人的特色, 像社会安全号, 是可以考虑的。个别被选择的标识符能被中央权威机构注册, 但是一个解决这种名字冲突的手段是必要的。

特殊注意点是为一个个体容纳多个名字的需要, 为了代表和允许个体可能扮演的多个角色代理。一个命名机制绑定用户到需要的密钥。绑定不可能是不变的因此角色有时改变(例如, 一个公司的审计员被解雇)。

检查扩展 DARPA/DoD 域名系统是适宜的并且它和名字服务器相连解决对于单独用户 ID 的用户名。一个额外的发行和邮件列表的支持一起出现: 名字服务器当前没有执行用户

列表的扩展（潜在递归的）。ISO 和 CSNet 正在研究用户级的路径服务机制，也可以进行考虑。

7. 用户接口和实现的例子

为了将在本 RFC 中讨论的机制和方法放入到设备环境中，这一节介绍了一个原型实现。这个实现是一个被用户调用的独立的程序，分散在存在的用户的子层。这样一个程序能被调用作为一个在电子邮件用户代理或文本编辑器内的过滤器，简化必须被用户操作的操作顺序。这个集成形式提供了程序和一定范围 UA 程序一起使用的优势，而不是仅仅和一个特殊的 UA 兼容。

当一个用户希望对一个发出的消息应用保密增强，用户准备消息的文本并调用单独的程序（和程序相互作用为了提供地址信息和其它进行保密增强处理需要的数据），依次产生适合通过 UA 传输的输出。当一个用户接收到一个保密增强消息，UA 以被加密的形式传送消息，适于被单独的程序解密和做相关处理。

在这个原型（基于对称密钥管理）IK 组件的存储被维护在一个本地的文件中，输入项基于发送者和接收者提供的信息手工管理。这个存储是一个有效简单的数据库。IK 组件被选择用于传送基于发送者识别和接收者名字的消息，相应的“X-Sender-ID:”和“X-Recipient-ID:”被放置在消息的头。当一个消息被接收，这些域被用作一个在数据库查循的基础，服从合适的 IK 组件入口。DEKs 和 IVs 在程序中动态产生。

选项和目的地址通过传给单独程序的命令行参数选择。规定对于保密增强邮件目的地址功能逻辑上不同于规定对应于到被 MTS 使用的 UA 的地址的功能。这一分别是由于在许多情况下被规定在一个 UA 中的一个地址的本地形式和使用在“X-Sender-ID:”和“X-Recipient-ID:”域中的互连网全球形式不同。

8. 进一步研究的领域

定义在本 RFC 中的过程足以支持在 Internet 互操作方的保密增强电子邮件传送的实现。将需要进一步的努力，然而，为了增强健壮性，一般性，和互操作性。特别以下的领域需要进一步研究：

1. 用户命名技术，和他们与域名系统，名字服务器，路径服务，和密钥管理功能的联系。
2. 发行机构和路径服务功能和交互的详细标准。
3. 和 X.400 邮件保密增强的互操作性。

我们期待以后的 RFC 文档将针对这些课题。

9. 参考

这一节标识了可能对于那些打算使用本文档中定义的机制有用的背景参考。

ISO 7498/Part 2 - Security Architecture, prepared by ISO/TC97/SC
21/WG 1 Ad hoc group on Security, extends the OSI Basic Reference

Model to cover security aspects which are general architectural elements of communications protocols, and provides an annex with tutorial and background information.

US Federal Information Processing Standards Publication (FIPS PUB) 46, Data Encryption Standard, 15 January 1977, defines the encipherment algorithm used for message text encryption and Message Authentication Code (MAC) computation.

FIPS PUB 81, DES Modes of Operation, 2 December 1980, defines specific modes in which the Data Encryption Standard algorithm may be used to perform encryption.

FIPS PUB 113, Computer Data Authentication, May 1985, defines a specific procedure for use of the Data Encryption Standard algorithm to compute a MAC.

注意:

- [1] Key generation for MIC computation and message text encryption may either be performed by the sending host or by a centralized server. This RFC does not constrain this design alternative. Section 5.1 identifies possible advantages of a centralized server approach if symmetric key management is employed.
- [2] American National Standard Data Encryption Algorithm (ANSI X3.92-1981), American National Standards Institute, Approved 30 December 1980.
- [3] Federal Information Processing Standards Publication 46, Data Encryption Standard, 15 January 1977.
- [4] Information Processing Systems: Data Encipherment: Modes of Operation of a 64-bit Block Cipher.
- [5] Federal Information Processing Standards Publication 81, DES Modes of Operation, 2 December 1980.
- [6] ANSI X9.17-1985, American National Standard, Financial Institution Key Management (Wholesale), American Bankers Association, April 4, 1985, Section 7.2.

- [7] Postel, J., "Simple Mail Transfer Protocol" RFC-821,
USC/Information Sciences Institute, August 1982.

- [8] This transformation should occur only at an SMTP endpoint, not at
an intervening relay, but may take place at a gateway system
linking the SMTP realm with other environments.

- [9] Use of the SMTP canonicalization procedure at this stage was
selected since it is widely used and implemented in the Internet
community, not because SMTP interoperability with this
intermediate result is required; no privacy-enhanced message will
be passed to SMTP for transmission directly from this step in the
four-phase transformation procedure.

- [10] Crocker, D., "Standard for the Format of ARPA Internet Text
Messages", RFC-822, August 1982.

- [11] Rose, M. and E. Stefferud, "Proposed Standard for Message
Encapsulation", RFC-934, January 1985.

- [12] CCITT Recommendation X.411 (1988), "Message Handling Systems:
Message Transfer System: Abstract Service Definition and
Procedures".

- [13] CCITT Recommendation X.509 (1988), "The Directory -
Authentication Framework".

- [14] Kille, S., "Mapping between X.400 and RFC-822", RFC-987, June
1986.

- [15] Federal Information Processing Standards Publication 113,
Computer Data Authentication, May 1985.

- [16] American National Standard for Information Systems - Data
Encryption Algorithm - Modes of Operation (ANSI X3.106-1983),
American National Standards Institute - Approved 16 May 1983.

- [17] Voydock, V. and S. Kent, "Security Mechanisms in High-Level
Network Protocols", ACM Computing Surveys, Vol. 15, No. 2, Pages
135-171, June 1983.

作者地址:

John Linn
Secure Systems
Digital Equipment Corporation
85 Swanson Road, BXB1-2/D04
Boxborough, MA 01719-1326

Phone: 508-264-5491

EMail: Linn@ultra.enet.dec.com