用户。用户A可从目录中得到相应的证书以建立到B的以下证书链：

$$X\ll W\gg W\ll V\gg V\ll Y\gg Y\ll Z\gg Z\ll B\gg$$

并通过该证书链获取B的公开钥。

类似地，B可建立以下证书链以获取A的公开钥：

$$Z\ll Y\gg Y\ll V\gg V\ll W\gg W\ll X\gg X\ll A\gg$$

3. 证书的吊销

每一证书都有一个有效期，然而有些证书还未到截止日期就会被发放该证书的CA吊销，这是由于用户的秘密钥有可能已被泄露，或者该用户不再由该CA来认证，或者CA为该用户签署证书的秘密钥有可能已泄露。为此，每一CA还必须维护一个证书吊销列表(Certificate Revocation List，CRL)，见图10-7(b)，其中存放所有未到期而被提前吊销的证书，包括该CA发放给用户和发放给其他CA的证书。CRL还必须经该CA签名，然后存放于目录以供他人查询。

CRL中的数据域包括发放者CA的名称、建立CRL的日期、计划公布下一CRL的日期以及每一被吊销的证书数据域，而被吊销的证书数据域包括该证书的顺序号和被吊销的日期。因为对一个CA来说，他发放的每一证书的顺序号是唯一的，所以可用顺序号来识别每一证书。

所以每一用户收到他人消息中的证书时，都必须通过目录检查这一证书是否已被吊销。为避免搜索目录引起的延迟以及由此而增加的费用，用户自己也可维护一个有效证书和被吊销证书的局部缓存区。

10.3.2 认证过程

X.509有3种认证过程以适应不同的应用环境。3种认证过程都使用公钥签名技术，并假定通信双方都可从目录服务器获取对方的公钥证书，或对方最初发来的消息中包括的公钥证书，即假定通信双方都知道对方的公钥。3种认证过程如图10-9所示。

1. 单向认证

单向认证指用户A将消息发往B，以向B证明：A的身份，消息是由A产生的，消息的意欲接收者是B，消息的完整性和新鲜性。

为实现单向认证，A发往B的消息应由A的秘密钥签署的若干数据项组成。数据项中应至少包括时间戳t_A、一次性随机数r_A、B的身份，其中时间戳又有消息的产生时间(可选项)和截止时间，以处理消息传送过程中可能出现的延迟，一次性随机数用于防止重放攻击。r_A在该消息未到截止时间以前应是这一消息唯一所有的，因此B可在这一消息的截止时间以前一直存有r_A，以拒绝具有相同r_A的其他消息。

如果仅为了单纯认证，则A发往B的上述消息就可作为A提交给B的凭证。如果不单纯为了认证，则A用自己的秘密钥签署的数据项还可包括其他信息sgnData，将这个信息也包括在A签署的数据项中可保证该信息的真实性和完整性。数据项中还可包括由B的公开钥PK_B加密的双方意欲建立的会话密钥K_{AB}。

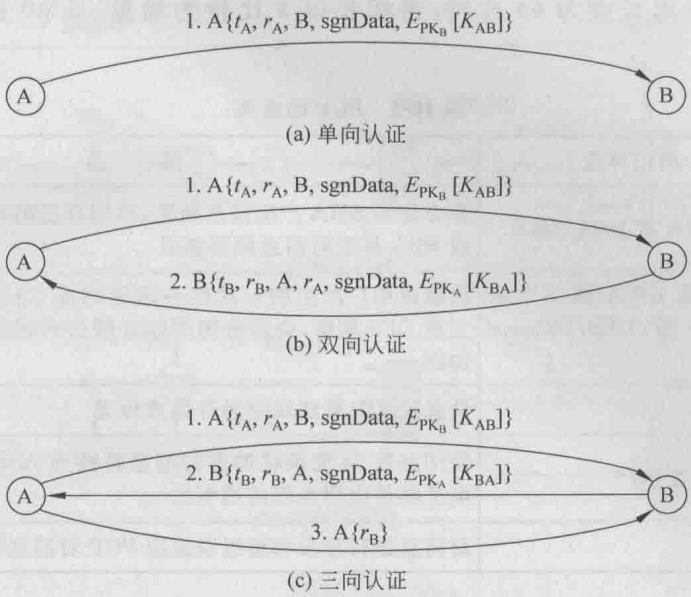


图 10-9　X.509 的认证过程

2. 双向认证

双向认证是在上述单向认证的基础上，B 再向 A 作出应答，以证明：B 的身份，应答消息是由 B 产生的，应答的意欲接收者是 A，应答消息是完整的和新鲜的。

应答消息中包括由 A 发来的一次性随机数 r_A（以使应答消息有效），由 B 产生的时间戳 t_B 和一次性随机数 r_B。与单向认证类似，应答消息中也可包括其他附加信息和由 A 的公开钥加密的会话密钥。

3. 三向认证

在上述双向认证完成后，A 再对从 B 发来的一次性随机数签名后发往 B，即构成第三向认证。三向认证的目的是双方将收到的对方发来的一次性随机数又都返回给对方，因此双方不需检查时间戳而只需检查对方的一次性随机数即可检查出是否有重放攻击。在通信双方无法建立时钟同步时，就需使用这种方法。

10.4　PGP

PGP(Pretty Good Privacy)是目前最为普遍使用的一种电子邮件系统。该系统能为电子邮件和文件存储应用过程提供认证业务和保密业务。

本节分别介绍 PGP 的运行方式、密钥的产生和存储以及公钥的管理。

10.4.1　运行方式

PGP 有 5 种业务：认证性、保密性、压缩、电子邮件的兼容性、分段。表 10-1 是这 5 种业务的总结。其中 CAST-128 是一种分组密码，算法具有传统 Feistel 网络结构，采用

16轮迭代,明文分组长度为64比特,密钥长以8比特为增量,从40比特到128比特可变。

表10-1 PGP的业务

功 能	所用算法	描 述
数字签名	DSS/SHA或RSA/SHA	发方使用SHA产生消息摘要,再用自己的秘密钥按DSS算法或RSA算法对消息摘要签名
消息加密	CAST或IDEA或三个密钥的三重DES/ElGamal或RSA	消息由用户产生的一次性会话密钥按CAST-128或IDEA或三重DES加密,会话密钥用收方的公开钥按ElGamal或RSA加密
压缩	ZIP	消息经ZIP算法压缩后存储或传送
电子邮件的兼容性	基数64变换	使用基数64变换将加密的消息转换为ASCII字符串,以提供电子邮件应用系统的透明性
分段		对消息进行分段和重组以适应PGP对消息最大长度的限制

图10-10是PGP的认证业务和保密业务示意图。其中,K_S为分组加密算法所用的会话密钥;EC和DC分别为分组加密算法和解密算法;EP和DP分别为公钥加密算法和解密算法;SK_A和PK_A分别为发方的秘密钥和公开钥;SK_B和PK_B分别为收方的秘密钥和公开钥;H表示哈希函数;‖表示链接;Z为ZIP压缩算法;$R64$表示基数64变换。

1. 认证业务

图10-10(a)表示PGP中通过数字签名提供认证的过程,分为5步:

(1) 发方产生消息M。

(2) 用SHA产生160比特长的消息摘要$H(M)$。

(3) 发方用自己的秘密钥SK_A按RSA算法对$H(M)$加密,并将加密结果$EP_{SK_A}[H(M)]$与M链接后发送。

(4) 收方用发方的公开钥对$EP_{SK_A}[H(M)]$解密得$H(M)$。

(5) 收方对收到的M计算消息摘要,并与(4)中的$H(M)$比较。如果一致,则认为M是真实的。

过程中结合使用了SHA和RSA算法,类似地也可结合使用DSS算法和SHA算法。

以上过程将消息的签名与消息链接后一起发送或存储,但在有些情况中却将消息的签名与消息分开发送或存储。例如,将可执行程序的签名分开存储,以后可用来检查程序是否有病毒感染。再如,多人签署同一文件(如法律合同),每人的签名都应与被签文件分开存放;否则,第一个人签完字后将消息与签名链接在一起,第二人签名时既要签消息,又要签第一人的签名,因此就形成了签名的嵌套。

2. 保密业务

PGP的另一业务是为传输或存储的文件提供加密的保密性业务。加密算法用CAST-128,也可用IDEA或三重DES,运行模式为64比特CFB模式。加密算法的密钥为一次性的,即每加密一个消息时都需产生一个新的密钥,称为一次性会话密钥,且新密

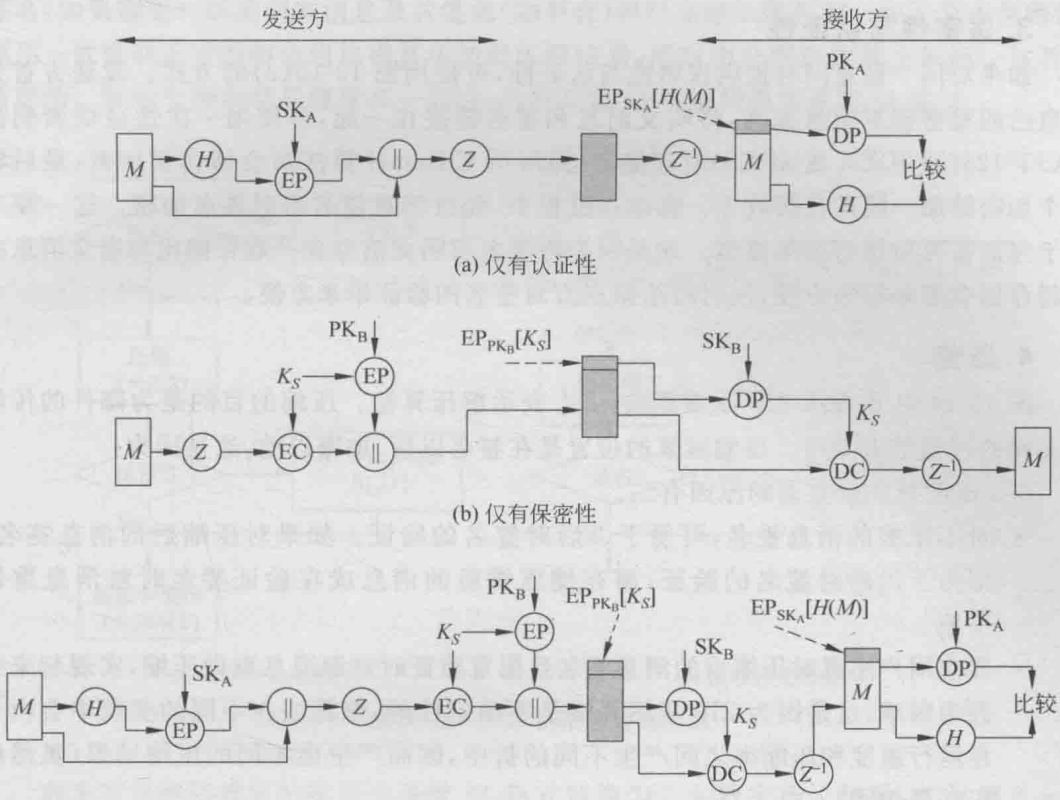


图 10-10　PGP 的认证业务和保密业务

钥也需用接收方的公开钥加密后与消息一起发往收方，整个过程如下[见图 10-10(b)]：

(1) 发送方产生消息 M 及一次性会话密钥 K_S。

(2) 用密钥 K_S 按 CAST-128(或 IDEA 或 3DES)加密 M。

(3) 用接收方的公开钥 PK_B 按 RSA 算法加密一次性会话密钥 K_S，将(2)、(3)中的两个加密结果链接起来发往收方。

(4) 接收方用自己的秘密钥按 RSA 算法恢复一次性会话密钥。

(5) 接收方用一次性会话密钥恢复发方发来的消息。

PGP 还为加密一次性会话密钥提供了 ElGamal 算法以供选用。

以上方案有以下几个优点：第一，由于分组加密速度远快于公钥加密速度，因此使用分组加密算法加密消息、使用公钥加密算法加密一次性会话密钥可比单纯使用公钥算法大大地减少加密时间。第二，因为会话密钥是一次性的，因此没有必要使用会话密钥的交换协议。同时，由于电子邮件的存储转发特性，也无法使用握手交换协议。本方案使用公钥加密算法来传送一次性会话密钥，保证了仅接收方能得到。第三，一次性会话密钥的使用进一步加强了本来就很强的分组加密算法，因此只要公钥加密算法是安全的，整个方案就是安全的。PGP 可允许用户选择的密钥长度范围为 768～3072 比特，而若使用 DSS，则其密钥限制为 1024 比特。

3. 保密性与认证性

如果对同一消息同时提供保密性与认证性,可使用图10-10(c)的方式。发送方首先用自己的秘密钥对消息签名,将明文消息和签名链接在一起,再使用一次性会话密钥按CAST-128(或IDEA或3DES)对其加密,同时用ElGamal算法对会话密钥加密,最后将两个加密结果一同发往接收方。在这一过程中,先对消息签名再对签名加密。这一顺序优于先加密再对加密结果签名。这是因为将签名与明文消息在一起存储比与密文消息在一起存储会带来很多方便,同时也给第三方对签名的验证带来方便。

4. 压缩

图10-10中Z表示ZIP压缩算法,Z^{-1}表示解压算法。压缩的目的是为邮件的传输或文件的存储节省空间。压缩运算的位置是在签名以后、加密以前,这是因为:

(1) 压缩前产生签名的原因有二。

- 对不压缩的消息签名,可便于以后对签名的验证。如果对压缩后的消息签名,则为了以后对签名的验证,需存储压缩后的消息或在验证签名时对消息重做压缩。
- 即使用户愿意对压缩后的消息签名且愿意验证时对原消息重做压缩,实现起来也极为困难,这是因为ZIP压缩算法是不确定性的,该算法在不同的实现中会由于在运行速度和压缩率之间产生不同的折中,因而产生出不同的压缩结果(虽然解压结果相同)。

(2) 对消息压缩后再进行加密可加强其安全性,这是因为消息压缩后比压缩前的冗余度要小,因此会使得密码分析更为困难。

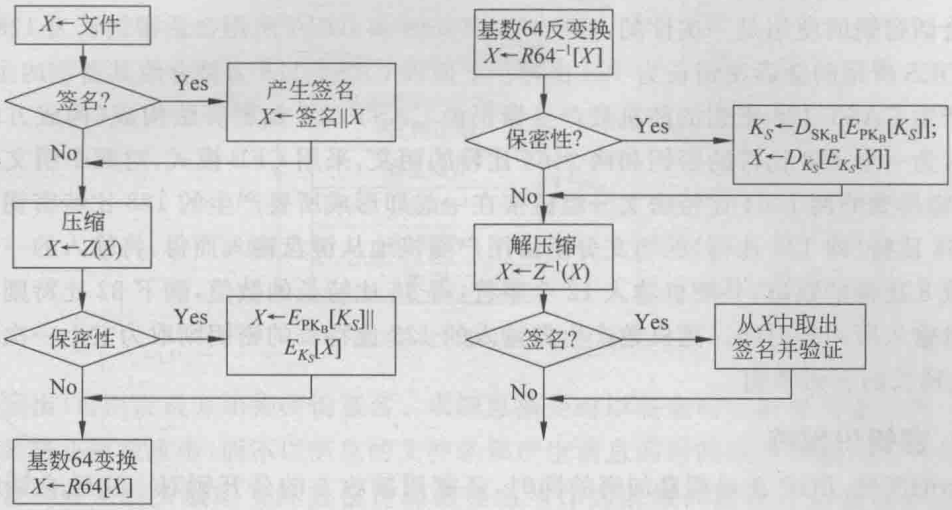
5. 电子邮件的兼容性

PGP在如图10-10所示的3种业务中,传输的消息都有被加密的部分(也许是所有部分),这些部分构成了任意8比特位组串。然而许多电子邮件系统只允许使用ASCII文本串,为此PGP提供了将8比特位串转换为可打印的ASCII字符的服务。转换方法是使用基数64变换,将每3个8比特位组的二元数据映射为4个ASCII字符。基数64变换可将被变换的消息扩展33%,但由于扩展是对会话密钥和消息的签名部分进行,而这一部分又是比较紧凑的,所以对明文消息的压缩足以弥补基数64变换所引起的扩展。有实例显示,ZIP的平均压缩率大约为2.0,因此如果不考虑相对小的签名和密钥部分,对长度为X的文件来说,压缩和扩展的总体效果为$1.33\times0.5\times X=0.665\times X$,即总体上有三分之一的压缩。

PGP变换具有"盲目性",即不管输入变换的消息内容是否是ASCII文件,都将变换为基数64格式。因此在图10-10所示的仅提供认证的服务中,对消息及其签名进行基数64变换,变换后的结果对不经意的观察者来说是不可读的,从而可提供一定程度的保密性。作为一种配置选择,PGP可以只将消息的签名部分转换为基数64格式,从而使得接收方不使用PGP就可阅读消息,但对签名的验证仍然需要使用PGP。

图10-11分别是发送方和接收方对消息的处理过程框图,发方首先对消息的哈希值

签名(如果需要),然后明文消息及其签名(如果有)再经压缩函数压缩。如果要求保密性,则用一次性会话密钥按分组加密算法加密压缩结果,同时用公钥加密算法加密一次性会话密钥。将两个加密结果链接在一起后,再经基数 64 变换转换为基数 64 格式。



(a) 发送方处理框图　　(b) 接收方处理框图

图 10-11　PGP 的消息处理框图

收方首先将接收到的结果由基数 64 格式转换为二元数字串。然后,如果消息是密文,则恢复一次性会话密钥,由一次性会话密钥恢复加密的消息,并对之解压缩。如果消息还经过签名,则从上一步恢复的消息中取出消息的哈希值,并与自己计算的消息的哈希值进行比较。

6. 分段与重组

电子邮件通常都对最大可用的消息长度有所限制,如果消息长度大于最大可用长度,则将消息分为若干子段并分别发送。分段是在图 10-11(a)基数 64 变换以后进行的,因此会话密钥报头和签名报头仅在第一子段的开头处出现一次。接收方在图 10-11(b)的处理过程以前,首先去掉第一子段开头处的报头再将各子段拼装在一起。

10.4.2　密钥和密钥环

PGP 所用的密钥有 4 类:分组加密算法所用的一次性会话密钥和基于密码短语的密钥,公钥加密算法所用的公开钥和秘密钥。为此 PGP 必须满足以下 3 个要求:

- 能够产生不可猜测的会话密钥。
- 用户可有多个公开钥-秘密钥对,这是因为用户可能希望随时更换自己的密钥对,另一方面用户可能希望在同一时间和多个通信方同时通信时分别使用不同的密钥对,或者用户可能希望通过限制一个密钥加密的内容的数量来增加安全性。所以用户和他的密钥对不是一一对应的关系,必须采取某一方式对密钥加以识别。

- PGP的每一用户都必须对存储自己密钥对的文件加以维护,同时还需对存储所有通信对方公钥的文件加以维护。

1. 会话密钥的产生

会话密钥的使用是一次性的,其中CAST-128和IDEA所用会话密钥长为128比特,三重DES所用的会话密钥长为168比特。下面以CAST-128为例介绍其密钥的生成。

产生CAST-128密钥的随机数产生器仍由CAST-128加密算法构成(构成方式略),其输入为一个128比特的密钥和两个64比特的明文,采用CFB模式,对两个明文分组加密,再将得到的两个64比特密文分组链接在一起即形成所要产生的128比特密钥。其中两个64比特(即128比特)的明文分组由用户随机地从键盘输入而得,将输入的一个字符表示成8比特的数值,共随机输入12个字符,得96比特长的数值,剩下32比特则用来表示键盘输入所用的时间。随机数产生器输入的128比特长的密钥则取为它上一次输出的128比特长的会话密钥。

2. 密钥识别符

如前所述,PGP在对消息加密的同时,还需用接收方的公开钥对一次性会话密钥加密,从而使得只有接收方能恢复会话密钥,进而恢复加密的消息。如果接收方只有一个密钥对(即公开钥-秘密钥对),就可直接恢复会话密钥。然而,接收方通常都有多个密钥对,他怎么知道会话密钥是用他的哪个公开钥加密的?一种解决办法是发送方将所用的接收方的公开钥一起发给接收方,但这种方法对空间的浪费太多,因为RSA的公开钥其长度可达数百位十进制数。另一种办法是对每一用户的每一公开钥都指定一个唯一的识别符,称为密钥ID,因此发送方用接收方的哪个公开钥就将这个公开钥的ID发给接收方。但使用这种方法必须考虑密钥ID的存储和管理,且收发双方都必须能够从密钥ID得到对应的公开钥,从而引起不必要的负担。PGP采用的方法是用公开钥中64个最低有效位表示该密钥的ID,即公开钥PK_A的ID是$PK_A \bmod 2^{64}$。由于64位已足够长,因而不同密钥的ID相重的概率非常小。

PGP在数字签名时也需对密钥加上识别符,这是因为发送方签名时可能有很多秘密钥可供使用,接收方必须知道使用发送方的哪一个公开钥来验证数字签名。PGP用签名中的64比特来表示相应公开钥的ID。

3. PGP的消息格式

图10-12表示PGP中发送方A发往接收方B的消息格式。其中,E_{PK_B}表示用接收方B的公开钥加密;E_{SK_A}表示用发送方A的秘密钥加密(即A的签名);E_{K_S}表示用一次性会话密钥K_S的加密;ZIP是压缩算法;$R64$是基数64变换。

PGP的消息由3部分组成:消息、消息的签名(可选)、会话密钥(可选)。

消息部分包括被存储或被发送的实际数据、文件名以及时间戳。

签名部分包括以下成分:

(1) 时间戳。产生签名的时间。

(2) 消息摘要。消息摘要是由SHA对签名的时间戳链接上消息本身后求得的160

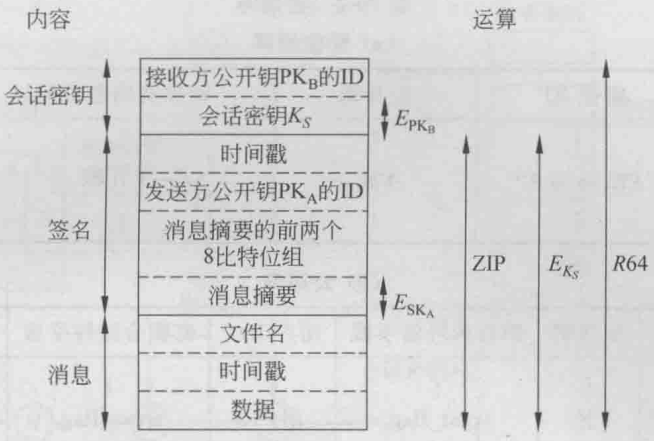


图 10-12　PGP 的消息格式

比特输出，再由发送方用秘密钥签名。求消息摘要时以签名时间戳作为输入的一部分的目的是防止重放攻击，而不以消息的文件名和产生消息的时间戳作为输入的一部分的目的是使得对无报头域的实际数据计算的签名与作为前缀而附加在消息前的签名完全一样。

(3) 消息摘要的前两个 8 比特位组：接收方用于与解密消息摘要后得到的前两个 8 比特位组进行比较，以确定自己在验证发送方的数字签名时是否正确地使用了发送方的公开钥。消息摘要的前两个 8 比特位组，也可用作消息的 16 比特帧校验序列。

(4) 发送方公开钥的 ID。用于标识解密消息摘要(即验证签名)的公开钥，相应地也标识了签名的秘密钥。

消息部分和签名部分经 ZIP 算法压缩后再用会话密钥加密。

会话密钥部分包括会话密钥和接收方公开钥标识符，标识符用于识别发送方加密会话密钥时使用接收方的哪个公开钥。

发送消息前，对整个消息作基数 64 变换。

4. 密钥环

为了有效存储、组织密钥，同时也为了便于用户的使用，PGP 为每个节点(即用户)都提供了两个表型数据结构：一个用于存储用户自己的密钥对(即公开钥/秘密钥)，另一个用于存储该用户所知道的其他各用户的公开钥。这两个数据结构分别称为秘密钥环和公钥环，如表 10-2 所示，其中带 * 的字段可作为标识字段。

在秘密钥环中，每行表示该用户的一个密钥对，其数据项有：产生密钥对的时间戳、密钥 ID、公开钥、被加密的秘密钥、用户 ID，其中密钥 ID 和用户 ID 可作为该行的标识符。用户 ID 可用用户的邮件地址，用户也可以为一个密钥对使用多个不同的用户 ID，还可以在不同的密钥对中使用相同的用户 ID。

秘密钥环由用户自己存储，仅供用户自己使用，而且为使秘密钥尽可能地安全，秘密钥是通过 CAST-128(或 IDEA 或 3DES)加密后以密文形式存储，加密过程为：用户首先

表 10-2 密钥环

(a) 秘密钥环

时间戳	密钥 ID*	公开钥	被加密的秘密钥	用户 ID*
⋮	⋮	⋮	⋮	⋮
T_i	$PK_i \bmod 2^{64}$	PK_i	$E_{H(P_i)}[SK_i]$	用户 i
⋮	⋮	⋮	⋮	⋮

(b) 公钥环

时间戳	密钥 ID*	公开钥	拥有者可信字段	用户 ID*	密钥合法性字段	签名	签名可信字段
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
T_i	$PK_i \bmod 2^{64}$	PK_i	$trust\_flag_i$	用户 i	$trust\_flag_i$		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

选择一个密码短语作为 SHA 的输入,产生出一个 160 比特的哈希值后销毁通行字短语,再用哈希值的 128 比特作为密钥按 CAST-128 对秘密钥加密,加密完成后再销毁哈希值。以后若要取出秘密钥,必须重新输入密码短语,PGP 产生出通行字的哈希值,并以此哈希值为秘密钥按 CAST-128 解密被加密的秘密钥。

从加密秘密钥的过程可见,秘密钥的安全性取决于所用密码的安全性,所以用户使用的密码应是易于自己记住的但又是不易被他人猜出的。

公钥环中每行存储的是该用户所知道的其他用户的公开钥,其数据项包括:时间戳(表示这一行产生的时间)、密钥 ID(指这一行的公开钥)、公开钥、用户 ID(指该公开钥的属主),其中密钥 ID 和用户 ID 可作为该行的标识符,还有其他几个数据项以后再介绍。

下面介绍消息传输和接收时密钥环是如何使用的。为简单起见,下面的过程中省略了压缩过程和基数 64 变换过程。

假定消息既要被签名,也要被加密,则发送方 A 需执行以下过程(见图 10-13,其中 RNG 是随机数产生器,其他符号和图 10-10 相同):

1) 签署消息

(1) PGP 使用 A 的用户 ID 作为索引(即关键字)从 A 的秘密钥环中取出 A 的秘密钥。如果用户 ID 为默认值,则从秘密钥环中取出第一个秘密钥。

(2) PGP 提示用户输入密码短语用于恢复被加密的秘密钥。

(3) 由 A 的秘密钥产生消息的签名。

2) 加密消息

(1) PGP 产生一个会话密钥,并由会话密钥对消息及签名加密。

(2) PGP 使用接收方 B 的用户 ID 作为关键字,从公钥环中取出 B 的公开钥。

(3) PGP 用 B 的公开钥加密会话密钥以形成发送消息中的会话密钥部分。

接收方 B 执行以下过程(见图 10-14):

1) 解密消息

(1) PGP 从接收到的消息中的会话密钥部分取出接收方 B 的密钥 ID,并以此作为关键字从 B 的秘密钥环中取出相应的被加密的秘密钥。

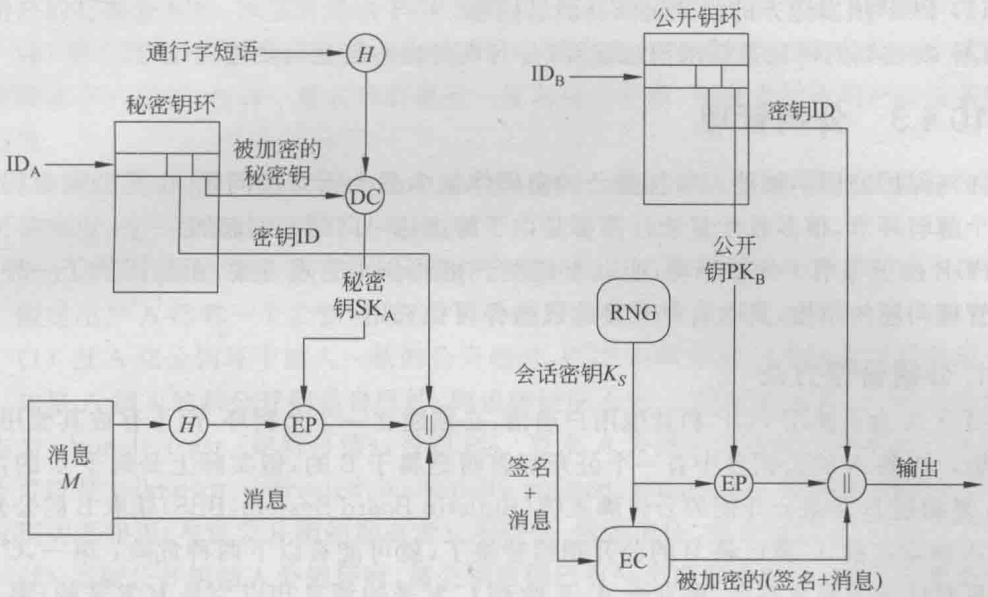


图 10-13　PGP 的消息产生过程

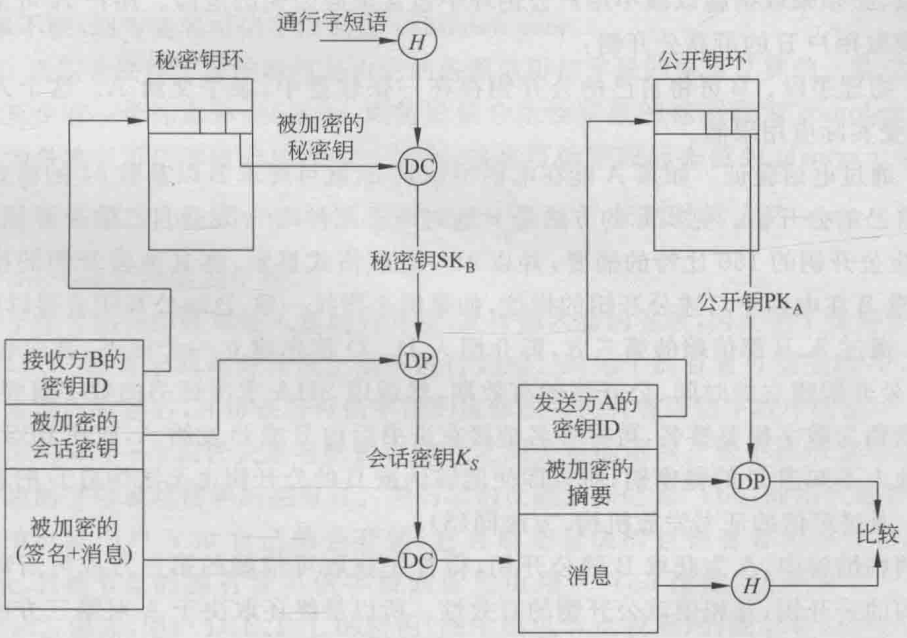


图 10-14　PGP 的消息接收过程

(2) PGP 提示 B 输入通行字短语以恢复秘密钥。

(3) PGP 用秘密钥恢复出会话密钥，并进而解密消息。

2) 认证消息

(1) PGP 从收到的消息中的签名部分取出发送方 A 的密钥 ID，并以此作为关键字从发送方的公钥环中取出发送方的公开钥。

(2) PGP用发送方的公开钥恢复消息摘要。

(3) 对收到的消息重新求消息摘要,并与恢复出的消息摘要进行比较。

10.4.3 公钥管理

如何保护公钥不被他人窜扰是公钥密码体制中最为困难的问题,也是公钥密码体制的一个薄弱环节,很多软件复杂性高都是由于解决这一问题而引起的。

PGP由于可用于各种环境,所以未建立严格的公钥管理方案,而是提供了一种解决公钥管理问题的结构,其中有好几种建议选择可供选用。

1. 公钥管理方法

用户A为了使用PGP和其他用户通信,必须建立一个公钥环,用于存放其他用户的公开钥。如果A的公钥环中有一个公开钥表明是属于B的,但实际上是属于C的,比如说,A是通过B发布公开钥的公告牌系统(Bulletin Board System,BBS)获取B的公开钥,但在A获取之前,C就已将B的公开钥给替换了,就可能有以下两种危险:第一,C可以假冒B对伪造的消息签名,再发给A,A收到C发来的消息却以为是B发来的;第二,A发给B的任何加密的消息都被C解读。

所以,必须采取措施以减小用户公钥环中包含虚假公钥的危险。用户A可采取以下措施以获取用户B的可靠公开钥:

(1) 物理手段。B可将自己的公开钥存在一张软盘中,亲手交给A。这个方法最为安全,但受实际应用限制。

(2) 通过电话验证。如果A能在电话中识别B,就可要求B以基数64的格式在电话中口述自己的公开钥。更实际的方法是B通过电子邮件向A发送自己的公开钥,A通过PGP产生公开钥的160比特的摘要,并以十六进制格式显示,称其为公开钥的指纹。然后A要求B在电话中口述公开钥的指纹,如果两个指纹一致,B的公开钥就得以验证。

(3) 通过A、B都信赖的第三方,即介绍人D。D首先建立一个证书,其中包括B的公开钥、公开钥建立的时间、公开钥的有效期,然后用SHA求出证书的数字摘要,并用自己的秘密钥为数字摘要签名,再将签名链接在证书后由B或D发给A,或在BBS中发布。因为其他人不知道D的秘密钥,所以即使能够伪造B的公开钥也无法伪造D的签名。

(4) 通过可信的证书发放机构,方法同(3)。

后两种措施中,A为获取B的公开钥,得首先获取可信赖的第三方或可信赖的证书发放机构的公开钥,并相信该公开钥的有效性。所以最终还取决于A对第三方或证书发放机构的信任程度。

2. PGP中的信任关系

PGP中虽然未对建立证书发放机构或建立可信赖机构做任何说明,但却提供了一种方便的方法来使用PGP中的信任关系,建立用户对公开钥的信任程度。

PGP建立信任关系的基本方法是用户在建立公钥环时,以一个公钥证书作为公钥环中的一行。其中有3个数据项用来表示对该公钥证书的信任程度(见表10-2):

(1) 密钥合法性字段(key legitimacy field):表示PGP以多大程度信任这一公开钥

是用户的有效公开钥，该字段是由 PGP 根据这一证书的签名可信字段计算的。

(2) 签名可信字段(signature trust field)：拥有该公开钥的用户可收集 0 个或多个为该公钥证书的签名，而每一签名后面都有一签名可信字段，用来表示该用户对签名者的信任程度。

(3) 拥有者可信字段(owner trust field)：用于表示用这一公开钥签署其他公钥证书的可信程度，这一字段由用户自己指定。

以上 3 个字段的取值是用来表示信任程度的标志。标志的具体含义在此不再赘述。

假定用户 A 已有一个公钥环，PGP 对公钥环的信任处理过程如下：

(1) 当 A 在公钥环中插入一新的公开钥时，PGP 必须为拥有者可信字段指定一个标志。如果 A 插入的新公开钥是自己的，则也将被插入到 A 的秘密钥环，PGP 自动地指定标志为 ultimate trust (绝对可信)；否则 PGP 要求 A 为该字段指定一个标志。A 指定的标志可以是 unknown、untrusted、marginally trusted、completely trusted 中的一个，其含义分别为不相识(与该公开钥的拥有者)、不信任、勉强信任、完全信任。

(2) 当新公开钥插入公钥环时，新公钥可能已有一个或多个签名，以后可能还会为这一公钥搜集多个签名。当为该公钥插入一新签名时，PGP 将在公钥环中查看签名产生者是否在已知的公钥拥有者中。如果在，则将拥有者可信字段中的标志赋值给签名可信字段；如果不在，则为签名可信字段赋值 unknown user。

(3) 密钥合法性字段的取值是由它的各签名可信字段的取值计算的。如果签名可信字段中至少有一个标志为 ultimate，则将密钥合法性字段的标志取为 complete；否则，为其赋值为各签名可信字段的加权和。其中，签名可信字段标志值为 always trusted 的权取为 $\frac{1}{X}$；标志值为 usually trusted 的权取为 $\frac{1}{Y}$；X、Y 分别是标志为 always trusted 和 usually trusted 的签名的个数。

由于在公钥环中既可插入新的公开钥，又可插入新的签名，因此为了保持公钥环中的一致性，PGP 都将定期地对其从上到下进行处理。对每个拥有者可信字段，PGP 将找出该拥有者的所有签名，并将签名可信字段的值修改为拥有者可信字段中的值。

图 10-15 是一个信任关系与密钥合法性之间关系的示例。图中圆节点表示密钥，圆节点旁边的字母表示密钥的拥有者。图的结构反映了标记为“You”的用户的公钥环。最顶层的节点是用户 You 自己的公开钥，它自然是合法的且拥有者可信字段的标志为 ultimate，其他节点的拥有者可信字段的标志由用户 You 指定，如果未指定则设置为 undefined。例如，用户 D、E、F、L 的密钥(图中黑色节点表示)的拥有者可信字段指定为 always trusts，意指 You 信任这 4 个密钥签署其他密钥，而用户 A、B 的密钥(图中灰色节点表示)的拥有者可信字段指定为 partially trusts，意即 You 部分信任这两个密钥签署其他密钥。

图的树结构表示哪个密钥已经被其他用户所签。例如，G 的密钥被用户 A 签署，则用一个由 G 指向 A 的箭头表示。如果一个密钥被一个其密钥不在公钥环中的用户签署，如 R，则用一个由 R 指向一个问号的箭头表示，问号表示签名者对用户 You 来说是未知的。

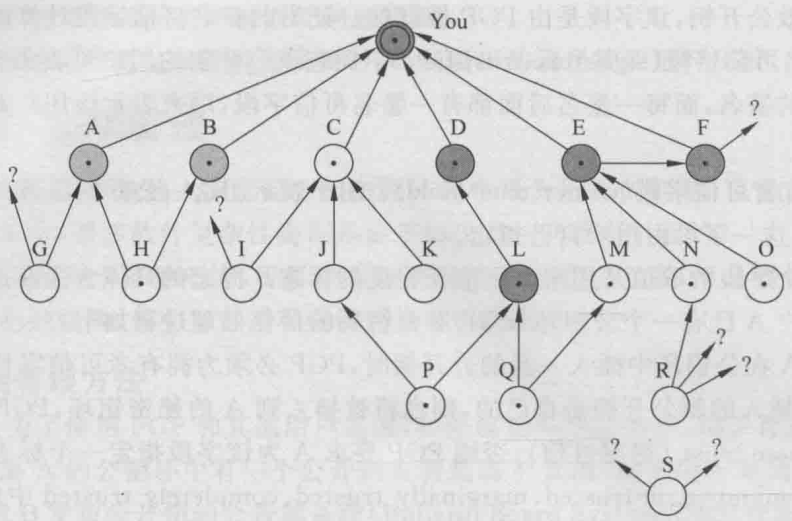


图 10-15 PGP 信任关系示例

该图可说明以下几个问题:

(1) 所有被 You 信任的用户(包括完全信任和部分信任)的公开钥都被 You 签署,节点 L 除外。这种签名并不总是有必要的。但实际中大多数用户都愿意为他们信任的用户的公开钥签名。例如,E 的公开钥已被 F 签署,但 You 仍直接为 E 的公开钥签名。

(2) 两个被部分信任的签名可用于证明其他密钥。例如,A 和 B 被 You 部分信任,则 PGP 认为 A 和 B 同时签署的 H 的公开钥是合法的。

(3) 由一个完全可信的或两个部分可信的签名者签署的公开钥被认为是合法的,但这一公开钥的拥有者可能不被信任签署其他公开钥。例如,由于 E 是 You 完全可信的,You 认为 E 为 N 签署的公开钥是合法的,但 N 却不被信任签署其他的公开钥。所以 R 的公开钥虽然被 N 签名,但 PGP 却不认为 R 的公开钥是合法的。这种情况具有实际意义。例如,如果想向某一用户发送一个保密消息,则不必在各方面都信任这一用户,只要确信这一用户的公开钥是正确的即可。

(4) 图中 S 是一个孤立节点,具有两个未知的签名者。这种公开钥可能是 You 从公钥服务器中得到的。PGP 不能由于这个公钥服务器的信誉好就简单地认为这个公开钥是合法的。

最后需指出的是,同一公开钥可能有多个用户 ID。这是因为用户可能改过自己的名字(即 ID)或者在多个 ID 名下申请了同一公开钥的签名,例如同一用户可能以自己的多个邮件地址作为自己的 ID 名。所以,可将具有多用户 ID 的同一公开钥以树结构组织起来,其中树根为这个公开钥,根的每一儿子是公开钥在一个 ID 下所获得的签名。对这个公开钥签署其他公开钥的信任程度取决于这一公开钥在不同 ID 下所获得的各个签名。

3. 公钥的吊销

用户如果怀疑与公开钥相应的秘密钥已被泄露,或者仅为避免使用同一密钥对的时间过长,就可吊销自己当前正使用的公开钥。这里所说的泄露指敌手从用户的秘密钥环

和密码短语恢复了用户经加密的秘密钥。吊销公开钥可通过发放一个经自己签名的公开钥吊销证书，证书的格式与前述公钥证书的格式一样，但多了一个标识符，用来表示该证书的目的是吊销这一公开钥。注意，用户签署公钥吊销证书的秘密钥是与被吊销的公开钥相对应的。用户还需尽可能快、尽可能广泛地散发自己的公开钥吊销证书，以使自己的各通信对方都尽可能快地更新自己的公钥环。

敌手如果得到用户的秘密钥也可以发放这种公钥吊销证书，从而使敌手自己和其他用户都不能继续使用这一公开钥。敌手以这种方式使用用户的秘密钥显然比以其他恶意方式使用用户的秘密钥所带来的危险小得多。

习　题

1. 下面是 X.509 三向认证的最初版本

$$A \rightarrow B: A\{t_A, r_A, B\}$$

$$B \rightarrow A: B\{t_B, r_B, A, r_A\}$$

$$A \rightarrow B: A\{r_B\}$$

假定协议不使用时间戳，可在其中将所有时间戳设置为 0，则攻击者 C 如果截获 A、B 以前执行协议时的消息，就可假冒 A 和 B，以使 A(B)相信通信的对方是 B(A)。请提出一种不使用时间戳的、防中间人欺骗的简单方法。

2. 在 PGP 中，若用户有 N 个公开钥，则密钥 ID 至少有两个重复的概率有多大？

3. 在 PGP 中，先对消息签名再对签名加密，请详细说明这一顺序为什么优于先加密再对加密结果签名。

4. 在 PGP 的消息格式中，消息摘要的前两个 8 比特位组以明文形式传输。

(1) 说明这种形式对哈希函数的安全性有多大程度的影响。

(2) 接收方用消息摘要的前两个 8 比特位组，确定自己在验证发送方的数字签名时是否正确地使用了发送方的公开钥，其可信程度有多大？

5. 在表 10-2 中，公钥环中的每一项都有一个拥有者可信字段，用于表示这一公钥的属主(即拥有这一公钥的用户)的可信程度。这一字段能否充分表达 PGP 对这一公钥的信任？如果不能，则如何实现 PGP 对这一公钥的信任？

参考文献

[1] 杨波. 网络安全理论与应用[M]. 北京：电子工业出版社，2002.

[2] William Stallings. Cryptography and Network Security: Principles and Practice. Second Edition [M]. Prentice Hall, New Jersey, 1999.

[3] 王育民，刘建伟. 通信网的安全——理论与技术[M]. 西安：西安电子科技大学出版社，1999.

[4] 卢开澄. 计算机密码学——计算机网络中的数据保密与安全[M]. 北京：清华大学出版社，1998.

[5] 朱文余，孙琦. 计算机密码应用基础[M]. 北京：科学出版社，2000.

[6] 卿斯汉. 密码学与计算机网络安全[M]. 北京：清华大学出版社，南宁：广西科学技术出版社，2001.

[7] 胡予濮，张玉清，肖国镇. 对称密码学[M]. 北京：机械工业出版社，2002.

[8] Arto Salomaa. Public-Key Cryptography, Second Edition[M]. Springer-Verlag, 1996.

[9] Hans Delfs, Helmut Knebl. Introduction to Cryptography[M]. Springer-Verlag, 2002.

[10] Jonathan Katz, Yehuda Lindell. Introduction to Modern Cryptography[M]. CRC Press, 2007.

[11] Wenbo Mao. Modern Cryptography: Theory and Practice[M]. Prentice Hall PTR, 2004.

[12] O Goldreich. Foundation of Cryptography: Basic Tools [M]. Cambridge University Press, Cambridge, 2001.

[13] O Goldreich. Foundation of Cryptography: Basic Applications[M]. Cambridge University Press, Cambridge, 2004.

[14] Joan Danmen, Vincent Rijmen. AES Proposal: Rijndael [M]. AES algorithm submission, September 3, 1999, AES home page: http://www.nist.gov/aes.

[15] D Boneh, M Franklin. Identity-based encryption from the Weil pairing [M]. SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003. Extended abstract in the Proceedings of Crypto 2001, LNCS 2139, pages 213-229, Springer-Verlag, 2001.

[16] E Fujisaki, T Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-Crypto '99, LNCS 1666, Springer-Verlag, pp. 537-554, 1999.

[17] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem [M]. ANTS-III, LNCS 1423, pp. 267-288, 1998. Springer-Verlag Berlin Heidelberg.

[18] Song Y Yan. Number Theory for Computing[M], Second, Springer-Verlag, 2002.

[19] D E Knuth. The Art of Computer Programming Ⅱ—Semi-numerical Algorithms [M], 3rd Edition, Addison-Wesley, 1998.

[20] S Goldwasser, S Micali. Probabilistic encryption[M]. Journal of Computer and System Sciences, 28: 270-299, 1984.

[21] M Naor, M Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks [M]. In 22nd Annual ACM Symposium on Theory of Computing, pages 427-437, 1990.

[22] C Racko, D Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack[M]. In Advances in Cryptology-Crypto'91, pages 433-444, 1991.

[23] ETSI/SAGE. Specification. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 1: 128-EEA3 & 128-EIA3 Specification. Version: 1.6, Date: 1st July 2011.

[24] ETSI/SAGE. Specification. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification. Version: 1.6, Date: 28th June 2011.