第 5 章 密钥管理技术

为了保护房间里的财物，人们通常在房门上安装一把锁，通过钥匙在锁孔里的旋转使锁的栓和结构按预定的方式形成障碍，以阻止房门被打开。开锁时，插入钥匙并反方向旋转使锁的栓和结构以相反的方式工作，从而将锁打开。密码学正是基于这一原理引入了“密钥”的概念。密钥作为密码变换的参数，起到“钥匙”的作用，通过加密变换操作，可以将明文变换为密文，或者通过解密变换操作，将密文恢复为明文。

在密码学中引入密钥的好处还表现为以下两方面：

(1) 在一个加密方案中不用担心算法的安全性，即可以认为算法是公开的，只要保护好密钥就可以了，很明显，保护密钥比保护算法要容易得多；

(2) 可以使用不同的密钥保护不同的秘密，这意味着当有人攻破了某个密钥时，受威胁的只是这个被攻破密钥所保护的信息，其他的秘密依然是安全的，由此可见密钥在整个密码算法中处于十分重要的中心地位。

密钥管理就是在授权各方之间实现密钥关系的建立和维护的一整套技术和程序。密钥管理是密码学的一个重要的分支，也是密码学应用中最重要、最困难的部分。密钥管理负责密钥从产生到最终销毁的整个过程，包括密钥的生成、存储、分发、使用、备份/恢复、更新、撤销和销毁等。密钥管理作为提供机密性、实体认证、数据源认证、数据完整性和数字签名等安全密码技术的基础，在整个密码学中占有重要的地位。因为现代密码学要求所有加密体制的密码算法是可以公开评估的，整个密码系统的安全性并不取决于对密码算法的保密或者对加密设备等的保护，而是取决于密钥的安全性，一旦密钥泄露，攻击者将有可能窃取到机密信息，也就不再具有保密功能。

密钥管理是一项综合性的系统工程，要求管理与技术并重，除了技术性因素外，它还与人的因素密切相关，包括与密钥管理相关的行政管理制度和密钥管理人员的素质。密码系统的安全强度总是取决于系统最薄弱的环节。因此，如果失去了必要管理的支持，再好的技术终将毫无意义，而管理只能通过健全相应的制度以及加强对人员的教育和培训来实现。

5.1 密钥管理的原则

密钥管理是一项系统工程，必须从整体上考虑，从细节处着手，进行严密、细致的施工，充分、完善的测试，才能较好地解决密钥管理问题。为此，首先应当弄清密钥管理的一些基本原则。

(1) 区分密钥管理的策略和机制。

密钥管理策略是密钥管理系统的高级指导。策略着重原则指导，而不着重具体实现。

密钥管理机制是实现和执行策略的技术和方法。没有好的管理策略，再好的机制也不能确保密钥的安全。反之，没有好的机制，再好的策略也没有实际意义。策略通常是原则性的、简单明确的，而机制是具体的、复杂烦琐的。

(2) 全程安全原则。

必须在密钥的产生、存储、备份、分发、组织、使用、更新、终止和销毁等全过程中对密钥采取妥善的安全管理。只有在各个环节都安全时，密钥才是安全的，否则只要其中一个环节不安全，密钥就不安全。例如对于重要的密钥，从它的产生到销毁的全过程中除了在使用的时候可以以明文形式出现外，其他环节中都不应当以明文形式出现。

(3) 最小权利原则。

应当只分发给用户进行某一事务处理所需的最小的密钥集合。因为用户获得的密钥越多，他的权利就越大，因而所能获得的信息就越多。如果用户不诚实，则可能发生危害信息安全的事件。

(4) 责任分离原则。

一个密钥应当专职一种功能，不要让一个密钥兼任几种功能。例如，用于数据加密的密钥不应同时用于认证，用于文件加密的密钥不应同时用于通信加密，而应当是：一个密钥用于数据加密，另一个密钥用于认证；一个密钥用于文件加密，另一个密钥用于通信加密。因为一个密钥专职一种功能，即使密钥泄露，也只会影响一种功能，使损失最小，否则损失就会大得多。

(5) 密钥分级原则。

对于一个大的系统(例如网络)，所需要的密钥的种类和数量都很多。应当采用密钥分级的策略，根据密钥的职责和重要性，把密钥划分为几个不同级别。用高级密钥保护低级密钥，最高级的密钥采用安全的物理保护。这样，既可减少受保护的密钥的数量，又可简化密钥的管理工作。一般可将密钥划分为三级：主密钥，二级密钥，初级密钥。

(6) 密钥更新原则。

密钥必须按时更新，否则，即使是采用很强大的密码算法，使用时间越长，攻击者截获的密文越多，破译密码的可能性就越大。理想情况是一个密钥只使用一次，完全的一次一密是不现实的。一般地，初级密钥采用一次一密，二级密钥更新的频率低些，主密钥更新的频率更低些。密钥更新的频率越高，越有利于安全，但是密钥的管理就越麻烦。实际应用时应当在安全和方便之间折中。

(7) 密钥应当有足够的长度。

密码安全的一个必要条件是密钥有足够的长度。密钥越长，密钥空间就越大，攻击就越困难，因而也就越安全。同时，密钥越长，软硬件实现消耗的资源就越多。

(8) 密码体制不同，密钥管理也不相同。

由于传统密码体制与公钥密码体制是性质不同的两种密码(例如，公钥密码体制的加密密钥可以公开，其秘密性不需要保护)，因此它们在密钥管理方法上有很大的不同。

5.2　密钥的层次结构

为了简化密钥管理工作，可采用密钥分级的策略。按照美国金融机构的密钥管理标准(ANSI X9.17)，将密钥分为初级密钥、密钥加密密钥和主密钥三个层次。

1. 初级密钥(Primary Key)

用于加解密数据的密钥称为初级密钥,也叫作数据加密密钥(Data Encrypting Key),它位于整个密钥层次体系的最底层。其中,直接用于为一次通信或数据交换中的用户数据提供保护的数据加密密钥称为会话密钥(Session Key),称用于文件保密的数据加密密钥为文件密钥(File Key)。

会话密钥一般由系统自动产生,且对用户是不可见的。一般来说,会话密钥只在会话存在期间有效,本次数据加、解密操作完成之后,会话密钥就将被立即清除,这是为了保证安全性。一方面,会话密钥变动越频繁,通信就越安全,因为攻击者所能获知的信息越少;另一方面,频繁改变的会话密钥将给其分发带来更多的负担;因此,作为系统安全策略的一部分,会话密钥生命期的确定需要进行折中考虑。而初级文件密钥与其所保护的文件有一样长的生存周期,一般要比初级会话密钥的生存周期长,有时甚至很长。

2. 密钥加密密钥(Key Encrypting Key)

密钥加密密钥是指用于对密钥进行加密操作的密钥,也称为二级密钥,它在整个密钥层次体系中位于初级密钥和主密钥之间,用于保护初级密钥。

3. 主密钥(Master Key)

主密钥是密钥层次体系中的最高级密钥,主密钥主要用于对密钥加密密钥进行保护。

系统使用主密钥通过某种算法保护密钥加密密钥,再使用密钥加密密钥通过算法保护会话密钥,不过密钥加密密钥可能不止一个层次,最后会话密钥基于某种加、解密算法来保护明文数据。在整个密钥层次体系中,各层密钥的使用由相应层次的密钥协议控制。

层次化的密钥结构意味着以少量高层密钥来保护大量下层密钥或明文数据,这样,可保证除了主密钥可以以明文的形式基于严格的管理受到严密保护外(不排除受到某种变换的保护),其他密钥则以加密后的密文形式存储,改善了密钥的安全性。

具体来说,层次化的密钥结构具有以下的优点。

(1) 安全性强。

一般情况下,位于层次化密钥结构中越下层的密钥更换得越快,最底层密钥可以做到每加密一份报文就更换一次。另外,在层次化的密钥结构中,下层的密钥被破译将不会影响到上层密钥的安全。在少量最初的、处于最高层次的主密钥注入系统之后,下面各层密钥的内容可以按照某种协议不断地变化(例如可以通过使用安全算法以及高层密钥产生低层密钥)。

对于破译者来说,层次化密钥结构意味着他所攻击的已不再是一个静止的密钥系统,而是一个动态的密钥系统。对于一个静态的密钥系统,一份报文被破译(得到加密该报文所使用的密钥)就会导致使用该密钥的所有报文的泄露;而在动态的密钥系统中,密钥处在不断的变化中,在底层密钥受到攻击后,高层密钥可以有效地保护底层密钥进行更换,从而最大限度地削弱了底层密钥被攻击所带来的影响,使得攻击者无法一劳永逸地破译密码系统,有效地保护了密钥系统整体的安全性。同时,一般来讲,直接攻击主密钥是很困难的,因为主密钥使用的次数比较有限,并且可能会受到严密的物理保护。

(2) 可实现密钥管理的自动化。

由于计算机的普及应用及飞速发展,计算机系统的信息量、计算机网络的通信量和用户数不断增长,对密钥的需求量也随之迅速增加,人工交换密钥已经无法满足需要,而且不能

实现电子商务等网络应用中双方并不相识的情况下的密钥交换，因此，必须解决密钥的自动化管理问题。

层次化密钥结构中，除了主密钥需要由人工装入以外，其他各层的密钥均可以设计由密钥管理系统按照某种协议进行自动地分发、更换、销毁等。密钥管理自动化不仅大大提高了工作效率，也提高了数据安全性。它可以使得核心的主密钥仅仅掌握在少数安全管理人员的手中，这些安全管理人员不会直接接触到用户所使用的密钥（由各层密钥进行自动地协商获得）与明文数据，而用户又不可能接触到安全管理人员所掌握的核心密钥，这样，核心密钥的扩散范围控制在最小，有助于保证密钥的安全性。

5.3 密钥的生命周期

密钥的管理贯穿在密钥的产生、存储、备份、分发、组织、使用、更新、终止和销毁等过程中。在本节简要介绍密钥的产生、存储、备份、终止和销毁，5.4节、5.5节再详细介绍密钥的分发。

5.3.1 密钥的产生

对于一个密码体制，如何产生好的密钥是很关键的，因为密钥的大小与产生机制直接影响密码系统的安全，好的密钥应当具有良好的随机性和密码特性，避免弱密钥的出现。密钥的生成一般都首先通过密钥生成器借助于某种噪声源（如光标和鼠标的位置、内存状态、按下的键、声音大小等）产生具有较好的统计分布特性的序列，然后再对这些序列进行各种随机性检验以确保其具有较好的密码特性。不同的密码体制其密钥的具体生成方法一般是不相同的，与相应的密码体制或标准相联系。

而且，不同级别的密钥产生的方式一般也不同。

（1）对于主密钥，虽然一般它的密钥量很小，但作为整个密码系统的核心，需要严格保证它的随机性，避免可预测性。因此，主密钥通常采用掷硬币、骰子或使用物理噪声发生器的方法来产生。

（2）密钥加密密钥可以采用伪随机数生成器、安全算法或电子学噪声源产生。

（3）初级密钥可以在密钥加密密钥的控制下通过安全算法动态地产生。

5.3.2 密钥的存储和备份

密钥的存储是密钥管理中的一个十分重要的环节，而且也是比较困难的一个环节。所谓密钥的安全存储就是要确保密钥在存储状态下的秘密性、真实性和完整性。安全可靠的存储介质是密钥安全存储的物质条件，安全严密的访问控制机制是密钥安全存储的管理条件。只有当这两个条件同时具备时，才能确保密钥的安全存储。密钥安全存储的原则是不允许密钥以明文形式出现在密钥管理设备之外。

不同级别的密钥应当采用不同的存储方式。

（1）主密钥的存储。

主密钥是最高级的密钥，主要用于对二级密钥和初级密钥进行保护。主密钥的安全性要求最高，而且生存周期很长，需要采取最安全的存储方法。

由于主密钥是最高级的密钥,所以它只能以明文形态存储,否则便不能工作。这就要求存储器必须是高度安全的,不但物理上是安全的,而且逻辑上也是安全的。通常将其存储在专用密码装置中。

(2) 密钥加密密钥的存储。

密钥加密密钥可以以明文形态存储,也可以以密文形态存储。但是,如果以明文形式存储,则要求存储器必须是高度安全的,最好是与主密钥一样存储在专用密码装置中。如果以密文形态存储,则对存储器的要求降低。通常采用以主密钥加密的形式存储密钥加密密钥,这样可减少明文形态密钥的数量,便于管理。

(3) 初级密钥的存储。

由于会话密钥按"一次一密"的方式工作,使用时动态产生,使用完毕后即销毁,生命周期很短。因此,初级会话密钥的存储空间是工作存储器,应当确保工作存储器的安全。而对于初级文件密钥,由于它的生命周期较长,因此它需要妥善的存储。初级文件密钥一般采用密文形态存储,通常采用以二级文件密钥加密的形式存储初级文件密钥。

为了进一步确保密钥和加密数据的安全,对密钥进行备份是必要的。目的是一旦密钥遭到毁坏,可利用备份的密钥恢复原来的密钥或被加密的数据,避免造成损失。密钥备份本质上也是一种存储。

密钥的备份是确保密钥和数据安全的一种有备无患的措施。备份的方式有多种,除了用户自己备份外,也可以交由可信任的第三方进行备份,还可以以密钥分量形态委托密钥托管机构备份或以门限方案进行密钥分量的分享方式备份。因为有了备份,所以在需要时可以恢复密钥,从而避免损失。

5.3.3 密钥的终止和销毁

密钥的终止和销毁同样是密钥管理中的重要环节,但是由于种种原因这一环节往往容易被忽视。

当密钥的使用期限到期时,必须终止使用该密钥,并更换新密钥。对终止使用的密钥一般并不要求立即销毁,而需要再保留一段时间后再销毁,这是为了确保受其保护的其他密钥和数据得以妥善处理。只要密钥尚未销毁,就必须对其进行保护,丝毫不能疏忽大意。密钥销毁要彻底清除密钥的一切存储形态和相关信息,使得重复这一密钥成为不可能。这里既包括处于产生、分发、存储和工作状态的密钥及相关信息,也包括处于备份状态的密钥和相关信息。

值得注意的是,要采用妥善的方法清除存储器。对于磁记录存储器,简单地删除、清0或写1都是不安全的。

5.4 密钥分发和密钥协商

密钥分发和密钥协商过程都是用以在保密通信的双方之间建立通信所使用密钥的安全协议(或机制)。在这种协议(或机制)运行结束时,参与协议运行的双方都将得到相同的密钥,同时,所得到的密钥对于其他任何方都是不可知的,当然,可能参与的可信管理机构(Trust Authority,TA)除外。

密钥分发与密钥协商过程具有如下的一些区别：密钥分发被定义为一种机制，保密通信中的一方利用此机制生成并选择秘密密钥，然后把该密钥发送给通信的另一方或通信的其他多方；而密钥协商则是保密通信双方(或更多方)通过公开信道的通信来共同形成秘密密钥的协议。一个密钥协商方案中，用来确定密钥的值是通信双方所提供的输入的函数。

在密钥分发与密钥协商过程中，经常利用一个可信管理机构，它的作用可能包括：验证用户身份；产生、选择和传送秘密密钥给用户；等等。

5.4.1　密钥分发

密钥分发也称为密钥分配，从分发途径的不同，秘密密钥的分发可分为离线分发和在线分发两种方式。

离线分发方式是通过非通信网络的可靠物理渠道携带密钥并分发给互相通信的各用户。但这种方式有很多缺点，主要包括：随着用户的增多和通信量的增大，密钥量大大增加；密钥的更新很麻烦；不能满足电子商务等网络应用中陌生人间的保密通信需求；密钥分发成本过高。

在线分发方式是通过通信与计算机网络对密钥在线、自动地进行分发，主要是通过建立一个密钥分发中心(KDC)来实现。在这种方式中，每个用户将与 KDC 共享一个保密的密钥，KDC 可以通过该密钥来鉴别某个用户，会话密钥将按请求由 KDC 进行传送。

在具体执行密钥交换时有两种处理方式。

1. 会话密钥由通信发起方生成

如图 5.1 所示，当 A 与 B 要进行保密通信时，A 临时随机地选择一个会话密钥 SK_{A-B}，并用它与 KDC 间的共享密钥 K_{A-KDC} 加密这个会话密钥和希望与之通信的对象 B 的身份后发送给 KDC。

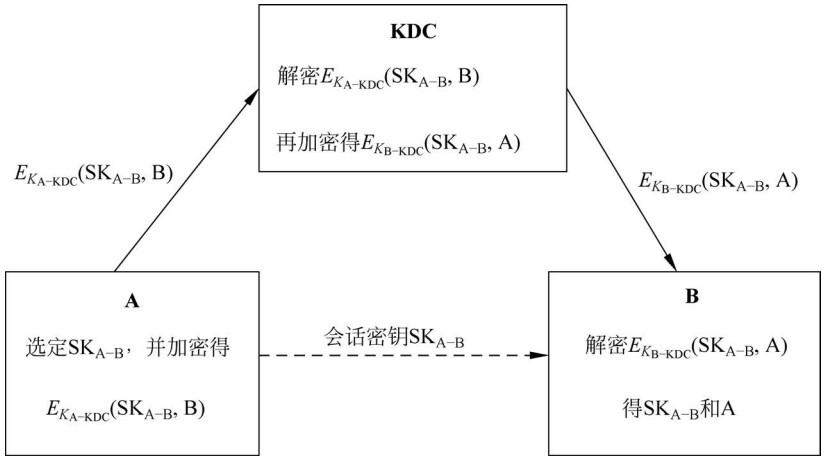


图 5.1　会话密钥由通信发起方生成

KDC 收到后再用 K_{A-KDC} 解密这个密文，获得 A 所选择的会话密钥 SK_{A-B} 以及 A 希望与之通信的对象 B，然后 KDC 用它和 B 之间共享的密钥 K_{B-KDC} 来加密这个会话密钥 SK_{A-B} 以及希望与 B 通信的对象 A 的身份，并将之发送给 B。

B 收到密文后，用它与 KDC 间共享的密钥 K_{B-KDC} 来解密，从而获知 A 要与自己通信

和 A 确定的会话密钥 $SK_{A\text{-}B}$。

然后,A 和 B 就可以用会话密钥 $SK_{A\text{-}B}$ 进行保密通信了。

2. 会话密钥由 KDC 生成

如图 5.2 所示,当 A 希望与 B 进行保密通信时,它先向 KDC 发送一条请求消息表明自己想与 B 通信。KDC 收到这个请求后就临时随机地产生一个会话密钥 $SK_{A\text{-}B}$,并将 B 的身份和所产生的这个会话密钥一起用 KDC 与 A 间共享的密钥 $K_{A\text{-}KDC}$ 加密后传送给 A。KDC 同时将 A 的身份和刚才所产生的会话密钥 $SK_{A\text{-}B}$ 用 KDC 与 B 间共享的密钥 $K_{B\text{-}KDC}$ 加密后传送给 B。

B 收到密文后,用它与 KDC 间共享的密钥 $K_{B\text{-}KDC}$ 来解密,从而获知 A 要与自己通信和 KDC 确定的会话密钥 $SK_{A\text{-}B}$。

然后,A 和 B 就可以用会话密钥 $SK_{A\text{-}B}$ 进行保密通信了。

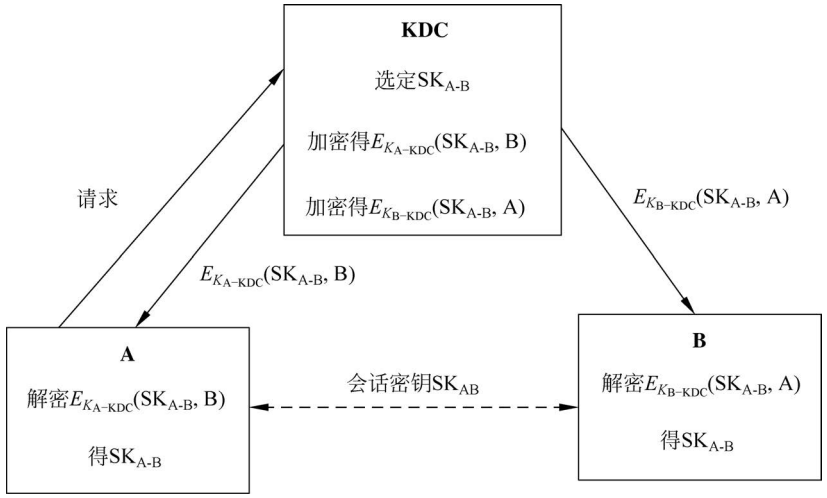


图 5.2 会话密钥由 KDC 生成

5.4.2 密钥协商

密钥协商是一个协议,它通过两个或多个成员在一个公开的信道上通信,共同建立一个秘密密钥。在一个密钥协商方案中,密钥的值是某个函数值,其输入量由成员提供。事实证明,实现密钥协商协议时采用公开密码效果最好。

本节中将介绍三种典型的密钥协商方案。

1. Diffie-Hellman 密钥协商协议

Diffie-Hellman 密钥协商协议是 Diffie 和 Hellman 在 1976 年提出的,它是第一个公开发表的公开密钥密码算法,其安全性基于有限域 Z_p 的离散对数问题,其算法描述如下:

设 p 是一个满足安全性要求的大素数,α 是模 p 的一个本原元。α 和 p 作为系统参数公开,所有的用户都可以得到 α 和 p,它们也将为所有的用户所共用。

在两个用户 A 与 B 要通信时,他们可以通过下列步骤协商通信时所使用的密钥。

S1:用户 A 选取一个随机数 r_A,$1 \leqslant r_A \leqslant p-2$,计算

$$s_A = \alpha^{r_A} \pmod{p}$$

并且把 s_A 发送给用户 B；

S2：用户 B 选取一个随机数 r_B，$1 \leqslant r_B \leqslant p-2$，计算

$$s_B = \alpha^{r_B} (\bmod p)$$

并且把 s_B 发送给用户 A；

S3：用户 A 计算 $K = s_B^{r_A} (\bmod p)$；

S4：用户 B 计算 $K' = s_A^{r_B} (\bmod p)$。

由于有

$$K = S_B^{r_A} (\bmod p) = (\alpha^{r_B} (\bmod p))^{r_A} (\bmod p)$$

$$= \alpha^{r_A r_B} (\bmod p) = s_A^{r_B} (\bmod p) = K'$$

这样，通信双方 A 与 B 得到共同的密钥 K，就可以使用对称密码体制进行保密通信。

如果攻击者窃听到了信道上的消息 s_A 与 s_B，由于 r_A 与 r_B 是用户在通信时才产生的保密参数，这样，如果想获得通信密钥 K 就必须求解离散对数问题，在素数很大时，这在计算上是不可行的。

【例 5.1】 已知密钥协商协议的公开参数素数 $p=97$ 和它的一个本原元 $\alpha=5$。A 和 B 分别选择随机数 $r_A=36$ 和 $r_B=58$。两人计算公开密钥如下：

$$A: s_A = 5^{36} \bmod 97 = 50 \bmod 97$$

$$B: s_B = 5^{58} \bmod 97 = 44 \bmod 97$$

在他们交换了公开密钥后，分别计算共享的会话密钥如下：

$$A: K = s_B^{r_A} (\bmod p) = 44^{36} \bmod 97 = 75 \bmod 97$$

$$B: K' = s_A^{r_B} (\bmod p) = 50^{58} \bmod 97 = 75 \bmod 97$$

可见，他们所得到并使用的会话密钥是一样的。

显然，Diffie-Hellman 密钥交换的基本模式为

$$\text{用户 A} \underset{\{\alpha^{r_B} (\bmod p)\}}{\overset{\{\alpha^{r_A} (\bmod p)\}}{\rightleftharpoons}} \text{用户 B}$$

但基本模式中，由于 Diffie-Hellman 协议没有提供身份鉴别，故容易遭受一个主动攻击者的中间人入侵攻击。主动攻击者 T 进行中间人攻击的基本模式为

$$\text{用户 A} \underset{\{\alpha^{r_B} (\bmod p)\}}{\overset{\{\alpha^{r_A} (\bmod p)\}}{\rightleftharpoons}} \text{攻击者 T} \overset{\{\alpha^{r'_A} (\bmod p)\}}{\rightleftharpoons} \text{用户 B}$$

主动攻击者 T 截取 A 与 B 之间的消息，且用自己的消息代替它们，图中 r'_A 和 r'_B 均为攻击者 T 产生。在协议结束时，用户 A 与攻击者 T 之间实际上形成了密钥 $\alpha^{r_A r'_B} (\bmod p)$，但误认为是与用户 B 共享的；用户 B 与攻击者 T 之间形成了密钥 $\alpha^{r'_A r_B} (\bmod p)$，但误认为是与用户 A 共享的。而用户 A、B 之间却并没有真正建立起共享密钥。当用户 A 要发送机密信息给用户 B 时，用户 A 用 $\alpha^{r_A r'_B} (\bmod p)$ 作为密钥加密信息，攻击者 T 截获并进行解密，然后把自己伪造的另一消息用密钥 $\alpha^{r_A r'_B} (\bmod p)$ 加密后，发送给用户 B。同样地，当用户 B 要发送加密消息给用户 A 时，情况类似。这样，攻击者 T 可成功欺骗用户 A、B，截获

并解密他们之间通信的机密信息,而用户A、B却不会发现被欺骗。

显然,用户A与用户B之间必须要有某种安全机制,使他们能够相互确认所形成的密钥真正是在两者之间形成,而不是与其他人(例如入侵者)共同建立的。这样,在密钥建立的同时,密钥协商协议应能同时鉴别参加者的身份,即身份鉴别过程必须与密钥协商过程紧密结合,以避免在身份鉴别后仍然有可能进行的中间人攻击,这种安全协议称为鉴别密钥协商协议。

因此在实际使用Diffie-Hellman协议时,必须引入某种鉴别机制,以使通信双方能够相互确认对方的身份,从而有效防止中间人攻击。

2. 端-端密钥协商协议

端-端(Stop-To-Stop,STS)协议是对Diffie-Hellman密钥交换协议进行安全加固后得到的密钥协商协议。该协议的基本特点是在Diffie-Hellman的密钥协商协议的基础上,通过引入数字签名机制来对抗中间人攻击。

该协议假定系统中每个用户都有一个数字签名方案,其中基于用户私钥的签名算法记为$\mathrm{Sig_A}$,基于用户公钥的验证算法记为$\mathrm{Ver_A}$。另外,系统中还有一个可信第三方TA,也有一个签名方案,签名算法为$\mathrm{Sig_{TA}}$,验证算法记为$\mathrm{Ver_{TA}}$。每个用户均持有由可信第三方TA颁发的数字证书,如用户A持有数字证书$C(\mathrm{A})$:

$$C(\mathrm{A})=(\mathrm{ID(A)},\mathrm{Ver_A},\mathrm{Sig_{TA}(ID(A)},\mathrm{Ver_A}))$$

其中,ID(A)标识了用户A的身份信息。

对简化的端-端密钥协商协议描述如下:

设p是一个满足安全性要求的大素数,α是模p的一个本原元。α和p作为系统参数公开。当系统中用户A与用户B要进行安全通信时,他们可以通过下列步骤协商通信时所使用的密钥。

S1:用户A选取一个随机数r_A,$1\leqslant r_\mathrm{A}\leqslant p-2$,计算$s_\mathrm{A}=\alpha^{r_\mathrm{A}}(\mathrm{mod}\ p)$,并把计算结果$s_\mathrm{A}$发送给用户B。

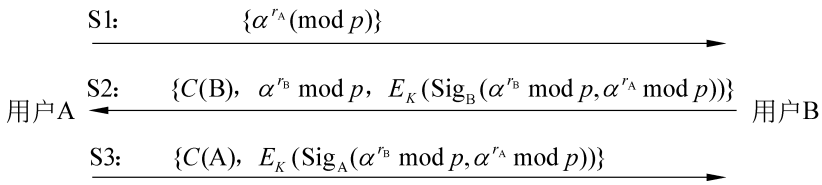
S2:用户B选取一个大的随机数r_B,$1\leqslant r_\mathrm{B}\leqslant p-2$,计算$s_\mathrm{B}=\alpha^{r_\mathrm{B}}(\mathrm{mod}\ p)$,接着计算$K=s_\mathrm{A}^{r_\mathrm{B}}(\mathrm{mod}\ p)$,和$e_\mathrm{B}=E_\mathrm{K}(\mathrm{Sig_B}(\alpha^{r_\mathrm{B}}\mathrm{mod}\ p,\alpha^{r_\mathrm{A}}\mathrm{mod}\ p))$,其中,$E_K(x)$表示以$K$为密钥对$x$进行对称加密,然后,用户B将$(C(\mathrm{B}),s_\mathrm{B},e_\mathrm{B})$发送给用户A。

S3:用户A计算$K=s_\mathrm{B}^{r_\mathrm{A}}(\mathrm{mod}\ p)$,并用$K$解密$e_\mathrm{B}$,得到用户B的数字签名$\mathrm{Sig_B}(\alpha^{r_\mathrm{B}}\mathrm{mod}\ p,\alpha^{r_\mathrm{A}}\mathrm{mod}\ p)$,然后,用户A用可信第三方TA的验证算法$\mathrm{Ver_{TA}}$验证用户B的证书$C(\mathrm{B})$的有效性,再用用户B的验证算法$\mathrm{Ver_B}$验证用户B的数字签名,如有任何一个验证失败则终止密钥协商,否则进入第4步(S4)。

S4:用户A计算$e_\mathrm{A}=E_K(\mathrm{Sig_A}(\alpha^{r_\mathrm{B}}\mathrm{mod}\ p,\alpha^{r_\mathrm{A}}\mathrm{mod}\ p))$,并把$(C(\mathrm{A}),e_\mathrm{A})$发送给用户B。

S5:用户B解密e_A后,得到用户A的数字签名$\mathrm{Sig_A}(\alpha^{r_\mathrm{A}},\alpha^{r_\mathrm{B}})$,使用可信第三方TA的验证算法$\mathrm{Ver_{TA}}$验证用户A的证书$C(\mathrm{A})$的有效性,使用用户A的验证算法$\mathrm{Ver_A}$验证用户A数字签名的有效性,如有任何一个验证失败则密钥协商失败。

在STS协议中,用户A与用户B之间的消息交换过程可以图解如下:



下面简要分析该协议是如何对抗中间人攻击的。如前所述，攻击者 T 截获步骤 S1 中用户 A 发给用户 B 的消息$\{\alpha^{r_A} (\mathrm{mod}\ p)\}$，并用$\{\alpha^{r'_A} (\mathrm{mod}\ p)\}$代替。之后 T 截获步骤 S2 中的消息$\{\mathrm{C(B)}, \alpha^{r_B} \bmod p, E_K(\mathrm{Sig_B}(\alpha^{r_B} \bmod p, \alpha^{r_A} \bmod p))\}$。若攻击者想用$\{\alpha^{r'_A} (\mathrm{mod}\ p)\}$来代替$\{\alpha^{r_A} (\mathrm{mod}\ p)\}$，就必须用 $\mathrm{Sig_B}(\alpha^{r_B} \bmod p, \alpha^{r'_A} \bmod p)$来代替 $\mathrm{Sig_B}(\alpha^{r_B} \bmod p, \alpha^{r_A} \bmod p)$，但他没有用户 B 的签名私钥，不能计算出$(\alpha^{r_B} \bmod p, \alpha^{r'_A} \bmod p)$的有效数字签名。

与此类似，因为攻击者没有用户 A 的签名私钥，从而也不能伪造用户 A 对$(\alpha^{r'_B} \bmod p, \alpha^{r_A} \bmod p)$的数字签名。

STS 协议通过数字签名验证了通信双方的身份，从而有效防止了中间人攻击。

3. 国密密钥协商协议 SM2-2

SM2-2 是国密算法 SM2 中的密钥交换协议，适用于商用密码应用中的密钥交换，可满足通信双方经过两次或可选三次信息传递过程，计算获取一个由双方共同决定的共享秘密密钥(会话密钥)。

(1) 符号和缩略语。

A，B	使用公钥密码系统的两个用户。
d_A	用户 A 的私钥。
d_B	用户 B 的私钥。
F_q	包含 q 个元素的有限域。
$E(F_q)$	F_q 上椭圆曲线 E 的所有有理点(包括无穷远点 O)组成的集合。
G	椭圆曲线的一个基点，其阶为素数。
Hash()	密码散列算法。
$H_v()$	消息摘要长度为 v 比特的密码散列算法。
h	余因子，$h = \sharp E(F_q)/n$，其中 n 是基点 G 的阶。
$\mathrm{ID_A}$，$\mathrm{ID_B}$	用户 A 和用户 B 的可辨别标识。
K，K_A，K_B	密钥交换协议商定的共享秘密密钥。
KDF()	密钥派生函数。
mod n	模 n 运算。例如，23 mod 7＝2。
n	基点 G 的阶(n 是 $\sharp E(F_q)$的素因子)。
O	椭圆曲线上的一个特殊点，称为无穷远点或零点，是椭圆曲线加法群的单位元。
P_A	用户 A 的公钥。
P_B	用户 B 的公钥。

q	有限域 F_q 中元素的数目。
a,b	F_q 中的元素,它们定义 F_q 上的一条椭圆曲线 E。
r_A	密钥交换中用户 A 产生的临时密钥值。
r_B	密钥交换中用户 B 产生的临时密钥值。
$x \| y$	x 与 y 的拼接,其中 x、y 可以是比特串或字节串。
Z_A	关于用户 A 的可辨别标识、部分椭圆曲线系统参数和用户 A 公钥的杂凑值。
Z_B	关于用户 B 的可辨别标识、部分椭圆曲线系统参数和用户 B 公钥的杂凑值。
$\#E(F_q)$	$E(F_q)$上点的数目,称为椭圆曲线 $E(F_q)$的阶。
$[k]P$	椭圆曲线上点 P 的 k 倍点,即$[k]P=P+P+\cdots+P$,k 是正整数。
$[x,y]$	大于或等于 x 且小于或等于 y 的整数的集合。
$\lceil x \rceil$	顶函数,表示大于或等于 x 的最小整数。例如,$\lceil 7 \rceil=7$,$\lceil 8.31 \rceil=9$。
$\lfloor x \rfloor$	底函数,表示小于或等于 x 的最大整数。例如,$\lfloor 7 \rfloor=7$,$\lfloor 8.3 \rfloor=8$。
&	两个整数的按比特与运算。

(2) 算法总体描述。

密钥交换协议是两个用户 A 和 B 通过交互的信息传递,用各自的私钥和对方的公钥来商定一个只有他们知道的秘密密钥。这个共享的秘密密钥通常用在某个对称密码算法中。该密钥交换协议能够用于密钥管理和协商。

(3) 椭圆曲线系统参数。

椭圆曲线系统参数包括有限域 F_q 的规模 q(当 $q=2^m$ 时,还包括元素表示法的标识和约化多项式);定义椭圆曲线 $E(F_q)$的方程的两个元素 a、$b \in F_q$;$E(F_q)$上的基点 $G=(x_G,y_G)(G \neq O)$,其中 x_G 和 y_G 是 F_q 中的两个元素;G 的阶 n 及其他可选项(如 n 的余因子 h 等)。

(4) 用户密钥对。

用户 A 的密钥对包括其私钥 d_A 和公钥 $P_A=[d_A]G=(x_A,y_A)$。用户 B 的密钥对包括其私钥 d_B 和公钥 $P_B=[d_B]G=(x_B,y_B)$。

(5) 辅助函数。

在本部分的椭圆曲线密钥交换协议中,涉及三类辅助函数:密码杂凑算法、密钥派生函数与随机数发生器。这三类辅助函数的强弱直接影响密钥交换协议的安全性。

(6) 用户其他信息。

用户 A 具有长度为 entlen_A 比特的可辨别标识 ID_A,记 ENTL_A 是由整数 entlen_A 转换而得的两个字节;用户 B 具有长度为 entlen_B 比特的可辨别标识 ID_B,记 ENTL_B 是由整数 entlen_B 转换而得的两字节。在本部分规定的椭圆曲线密钥交换协议中,参与密钥协商的 A、B 双方都需要用密码杂凑算法求得用户 A 的杂凑值 Z_A 和用户 B 的杂凑值 Z_B。在本部分规定的椭圆曲线数字签名算法中,签名者和验证者都需要用密码杂凑算法求得用户 A 的杂凑值 Z_A 和用户 B 的杂凑值 Z_B。

将椭圆曲线方程的参数 a、b,G 的坐标 x_G、y_G 和 P_A 的坐标 x_A、y_A 的数据类型转换为比特串,$Z_A=H_{256}(\text{ENTL}_A \| \text{ID}_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$。

将椭圆曲线方程参数 a、b，G 的坐标 x_G、y_G 和 P_B 的坐标 x_B、y_B 的数据类型转换为比特串，$Z_B=H_{256}(\mathrm{ENTL_B}\|\mathrm{ID_B}\|a\|b\|x_G\|y_G\|x_B\|y_B)$。

(7) 密钥交换协议及流程。

设用户 A 和 B 协商获得的密钥数据的长度为 klen 比特，用户 A 为发起方，用户 B 为响应方。用户 A 和 B 双方为了获得相同的密钥，应实现如下运算步骤。

记 $w=\lceil(\lceil\log_2(n)\rceil/2)\rceil-1$。

用户 A：

A1：用随机数发生器随机产生 $r_A\in[1,n-1]$；

A2：计算椭圆曲线点 $R_A=[r_A]G=(x_1,y_1)$；

A3：将 R_A 发送给用户 B；

用户 B：

B1：用随机数发生器随机产生 $r_B\in[1,n-1]$；

B2：计算椭圆曲线点 $R_B=[r_B]G=(x_2,y_2)$；

B3：从 R_B 中取出域元素 x_2，将 x_2 的数据类型转换为整数，计算 $\bar{x}_2=2^w+(x_2\&(2^w-1))$；

B4：计算 $t_B=(d_B+\bar{x}_2\cdot r_B)\bmod n$；

B5：验证 R_A 是否满足椭圆曲线方程，若不满足则协商失败；否则从 R_A 中取出域元素 x_1，将 x_1 的数据类型转换为整数，计算 $\bar{x}_1=2^w+(x_1\&(2^w-1))$；

B6：计算椭圆曲线点 $V=[h\cdot t_B](P_A+[\bar{x}_1]R_A)=(x_V,y_V)$，若 V 是无穷远点，则 B 协商失败；否则将 x_V，y_V 的数据类型转换为比特串；

B7：计算 $K_B=\mathrm{KDF}(x_V\|y_V\|Z_A\|Z_B,\mathrm{klen})$；

B8：(选项) 将 R_A 的坐标 x_1、y_1 和 R_B 的坐标 x_2、y_2 的数据类型转换为比特串，计算 $S_B=\mathrm{Hash}(0\mathrm{x}02\|y_v\|\mathrm{Hash}(x_v\|Z_A\|Z_B\|x_1\|y_1\|x_2\|y_2))$；

B9：将 R_B、(选项 S_B) 发送给用户 A；

用户 A：

A4：从 R_A 中取出域元素 x_1，将 x_1 的数据类型转换为整数，计算 $\bar{x}_1=2^w+(x_1\&(2^w-1))$；

A5：计算 $t_A=(d_A+\bar{x}_1\cdot r_A)\bmod n$；

A6：验证 R_B 是否满足椭圆曲线方程，若不满足则协商失败；否则从 R_B 中取出域元素 x_2，将 x_2 的数据类型转换为整数，计算 $\bar{x}_2=2^w+(x_2\&(2^w-1))$；

A7：计算椭圆曲线点 $U=[h\cdot t_A](P_B+[\bar{x}_2]R_B)=(x_U,y_U)$，若 U 是无穷远点，则 A 协商失败；否则将 x_U、y_U 的数据类型转换为比特串；

A8：计算 $K_A=\mathrm{KDF}(x_U\|y_U\|Z_A\|Z_B,\mathrm{klen})$；

A9：(选项) 将 R_A 的坐标 x_1、y_1 和 R_B 的坐标 x_2、y_2 的数据类型转换为比特串，计算 $S_1=\mathrm{Hash}(0\mathrm{x}02\|y_U\|\mathrm{Hash}(y_U\|Z_A\|Z_B\|x_1\|y_1\|x_2\|y_2))$，并检验 $S_1=S_B$ 是否成立，若等式不成立，则从 B 到 A 的密钥确认失败；

A10：(选项)计算 $S_A=\mathrm{Hash}(0\mathrm{x}03\|y_U\|\mathrm{Hash}(x_U\|Z_A\|Z_B\|x_1\|y_1\|x_2\|y_2))$，并将 S_A 发送给用户 B。

用户 B：

B10：(选项)计算 $S_2=\mathrm{Hash}(0\mathrm{x}03\|y_v\|\mathrm{Hash}(x_v\|Z_A\|Z_B\|x_1\|y_1\|x_2\|y_2))$，

并检验 $S_2=S_A$ 是否成立,若等式不成立则从 A 到 B 的密钥确认失败。

注:如果 Z_A、Z_B 不是用户 A 和 B 所对应的散列值,则自然不能达成一致的共享秘密值。密钥交换协议流程如图 5.3 所示。

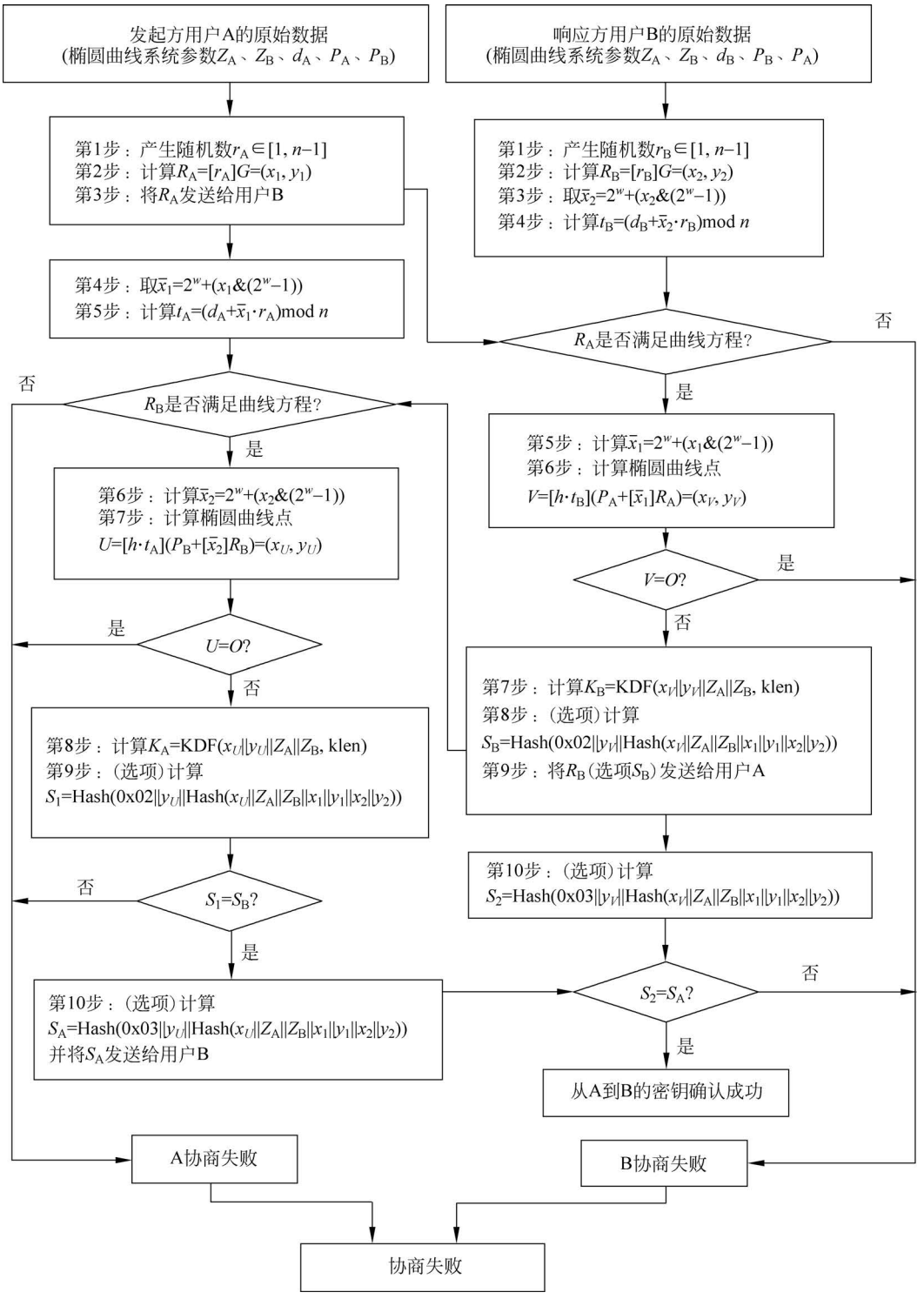


图 5.3 密钥协商协议流程

5.5　公开密钥的分发

5.5.1　公开密钥的分发方式

和传统密码体制一样，公开密钥密码体制在应用时也需要首先进行密钥分发。但是，公开密钥密码体制的密钥分发与传统密码体制的密钥分发有着本质的差别。由于传统密码体制中只要一个密钥，因此在密钥分发中必须同时确保密钥的秘密性、真实性和完整性。而公开密钥密码体制中有两个密钥，在密钥分发时必须确保用于解密或数字签名的私钥的秘密性、真实性和完整性，在实际应用中，私钥通常不需要传输，一般由使用者自己安全存储和保管。同时，因为加密密钥或签名验证密钥是公开的，因此分发公钥时，不需要确保其秘密性，但是却必须确保其真实性和完整性，绝对不允许攻击者替换或篡改用户的公钥。可以说，公开密钥密码体制中密钥管理的主要任务正是公钥的安全分发和安全管理。

归纳起来，公钥的分发方法主要有以下4种。

1）公开发布

公钥密码体制出现的原因之一是为了解决密钥分发问题，也就是公钥密码体制中的公钥可以公开，任一通信方可以将他的公钥发送给另一通信方或广播给其他通信各方。这种方法的突出优点是简便，密钥的分发不需要特别的安全渠道，相应地降低密钥管理的要求和成本；缺点是不能确保公钥的真实性和完整性，可能出现伪造公钥的公开发布，即某个用户可以假冒别的用户发送或广播公钥。

2）建立公钥目录

维护一个动态可访问的公钥目录（目录就是一组维护着用户信息的数据库）可以获得更大的安全性。这种方法要求一个称为目录管理员的可信实体或组织负责这个公开目录的维护，其他用户可以基于公开渠道访问该公钥目录来获取公钥。

这种方法比由个人公开发布公钥更安全，但也存在严重的缺点：一旦攻击者获知目录管理员的私钥，则他可以假冒任何通信方，传递伪造的公钥，或者修改目录管理员保存的记录，从而窃取发送给该通信方的消息；用户必须要知道这个公开目录的路径且信任该目录。

3）带认证的公钥分发（在线服务器方式）

一个更安全的公钥分发方法是在建立公钥目录的基础上增加认证功能：目录管理员负责维护通信各方公钥的动态目录，每个通信方可靠地知道该目录管理员的公钥，并且只有管理员知道对应的私钥。管理员向请求方返回经自己的私钥签名的被查找用户的公钥，请求方可以用管理员的公钥对所获取的签名变换过的公钥进行验证，从而确定该公钥的真实性。实际上，这是一种在线服务器式公钥分发解决方案。这种方案的处理步骤如下：

（1）A发送一条带有时间戳的消息给公钥目录管理员M，以请求B的当前公钥；

（2）M给A发送一条用其私钥 KR_M 签名的包括B的公钥 KR_B 在内的消息，这样A就可以用M的公钥对接收到的消息验证，因此A可以确信该消息来自M；

（3）A保存B的公钥，并用它对包含A的标识和及时交互号的消息加密，然后发送给B，其中及时交互号用来唯一标识本次信息交换；

（4）与A获取B的公钥一样，B以同样的方法从M处检索出A的公钥。

该方案的缺点是：只要用户与其他用户通信，就必须向目录管理员申请对方的公钥，故可信服务器必须在线，这可能导致可信服务器成为一个瓶颈。

4) 使用数字证书的公钥分发(离线服务器方式)

为了克服在线服务器式公钥分发解决方案的缺点，通信各方可以使用证书来交换密钥，获得良好的可靠性和灵活性。采用数字签名技术来确保公钥的真实性和完整性：将用户的标识符和用户的公钥联系在一起签名，则将用户的标识符和用户的公钥绑定在一起。如果破坏了这种对应关系，验证签名时将能够发现篡改行为。由此可见，采用数字签名技术可以确保公钥的安全分发。这里经过签名的一组信息的集合被称为证书，可信实体 X 被称为证书机构(Certification Authority，CA)。

一般地讲，证书是一个数据结构，是一种由一个可信任的管理机构签署的信息集合。在不同的应用中有不同的证书，如公钥证书、PGP 证书、SET 证书等，但这里只讨论公钥证书。

公钥证书是一种包含持证主体标识、持证主体公钥等信息，并由可信任的 CA 签署的信息集合。公钥证书主要用于确保公钥及其与用户绑定关系的安全。公钥证书的持证主体可以是人、设备、组织机构或其他主体。公钥证书的内容主要包括用户的名称、用户的公钥、证书的有效日期和 CA 的签名等。公钥证书能以明文的形式进行存储和分发。任何一个用户只要知道签证机构的公钥，就能检查对证书的签名的合法性。如果检查正确，那么用户就可以相信那个证书所携带的公钥是真实的，而且这个公钥就是证书所标识的那个主体的合法的公钥。

有了公钥证书系统后，如果某个用户需要任何其他已向 CA 注册的用户的公钥，可向持证人(或证书机构)直接索取其公钥证书，并用 CA 的公钥验证 CA 的签名，从而获得可信的公钥。由于公钥证书不需要保密，可以在互联网上分发，因而实现公钥的安全分发。又由于公钥证书有 CA 的签名，攻击者不能伪造合法的公钥证书。因此，只要 CA 是可信的，公钥证书就是可信的。其中 CA 公钥的获得也是通过证书方式进行的，为此 CA 也为自己颁发公钥证书。

使用公钥证书的主要好处是，用户只要获得 CA 的公钥，就可以安全地获得其他用户的公钥。因此公钥证书为公钥的分发奠定了基础，成为公开密钥密码在大型网络系统中应用的关键技术。这就是电子政务、电子商务等大型网络应用系统都采用公钥证书技术的原因。

5.5.2 X.509 公钥证书

目前应用最广泛的证书格式是国际电信联盟(International Telecommunication Union，ITU)提出的 X.509 版本 3 格式。X.509 标准最早于 1988 年颁布，在此之后又于 1993 年和 1995 年进行过两次修改。Internet 工程任务组(IETF)针对 X.509 在 Internet 环境的应用，颁布了一个作为 X.509 子集的 RFC 2459，从而使 X.509 在 Internet 环境中得到广泛应用。

1. X.509 证书的结构

X.509 版本 3 的证书结构如图 5.4 所示。

证书中各字段的说明如下。

版本号：证书的版本号。

证书序列号：由证书颁发者分发的本证书的唯一标识号。

签名算法标识符：由对象标识符加上相关参数组成，用于说明本证书所用的数字签名算法。例如，SHA-1 with RSA 说明该数字签名是利用 RSA 算法对数据的 SHA-1 的散列值进行签名。

颁发者名称：证书颁发者的可识别名。

有效期：证书有效的时间段。本字段由“起始时间”和“终止时间”两项组成，共同标识出证书的有效期。

主体名称：证书拥有者的可识别名。此字段必须是非空的，除非在扩展项中使用了其他名字形式。

主体公钥信息：主体的公钥以及所采用的公开密钥密码算法的标识符，这是必须说明的。

颁发者唯一标识符：证书颁发者的唯一标识符，属于可选字段，很少使用。

版本号
证书序列号
签名算法标识符
颁发者名称
有效期
主体名称
主体公钥信息（算法标识、公钥值）
颁发者唯一标识符（可选）
主体唯一标识符（可选）
扩展项（可选）
颁发者签名

图 5.4　X.509 版本 3 的证书结构

主体唯一标识符：证书拥有者的唯一标识符，属于可选字段，很少使用。

扩展项：在颁布了 X.509 版本 2 后，人们认为还有一些不足之处，于是提出一些扩展项附在版本 3 证书结构的后面。这些扩展项包括：密钥和策略信息、主体和颁发者属性以及证书路径限制。包含在扩展项中的信息可以标记为关键的，也可以标记为非关键的。如果某个扩展项标记为关键的，而应用程序又不能识别该扩展项类型，则应用程序应当拒绝该证书。如果某个扩展项标记为非关键的，即使应用程序不能识别该扩展项类型，应用程序也可以安全地忽略该扩展项并使用该证书。

颁发者签名：用证书颁发者私钥对证书其他字段生成的数字签名，以确保这个证书在颁发之后没有被篡改过。

2. 数字证书的管理

数字证书的管理包括证书的签署、发放、更新、查询以及作废等。

1）数字证书的签发

证书机构（说明见 5.6.1 节）接收、验证用户（包括下级证书机构和最终用户）数字证书的申请，将申请的内容进行备案，并根据申请的内容确定是否受理该数字证书申请。如果证书机构接受该数字证书的申请，则需进一步确定所颁发证书的类型。新证书用证书机构的私钥签名以后，发送到目录服务器供用户下载和查询。为了保证消息的完整性，返回给用户的所有应答信息都要使用证书机构的签名。

由于数字证书不需要特别的安全保护，因而可以使用不安全的协议在不可信的系统和网络上进行分发。

2）数字证书的更新

证书机构可以定期更新所有用户的证书，或者根据用户请求来更新特定用户的证书。

3）数字证书的查询

证书的查询可以分为证书申请的查询和用户证书的查询。前者要求证书机构根据用户的查询请求返回申请的处理过程和结果；后者则由目录服务器根据用户的请求返回适当的证书。

4) 数字证书的作废

数字证书都有一定的使用期限,其使用期限是由包含在证书中的开始日期和到期日期确定的,而有效期的长短(几个月至几年)则由证书机构的安全策略决定。

在某些特殊情况下(如对应的私钥已经失密、名字更改、主体与证书机构的关系已经改变等)要求提前作废该证书,证书机构通过周期性地发布并维护证书撤销列表(Certificate Revocation List,CRL)来完成上述功能。证书撤销列表是一个带时间戳的已作废数字证书列表,该列表由证书机构进行签名处理,供证书的使用者访问。CRL 可以发布在一个知名的互联网站上或是在证书机构自己的目录上进行分发。每份被撤销的证书在 CRL 中以其证书序列号进行标识。一个系统在使用一个经过认证的公钥前,不仅需要检查证书的签名和有效期,还要查询最新的 CRL 以确认该证书的序列号不在其中。因此,只有同时满足以下 3 个条件的证书才是有效的证书,即证书机构的签名有效、证书在有效期内,并且未被列入 CRL 中。

需要指出的是,随着电子商务的不断发展,SET 协议已暴露出一些问题和局限。例如,SET 协议本身比较复杂,较多的信息交互和计算代价降低了协议效率和便捷性;SET 协议更利于商家,协议没有规定收单银行给商家付款前必须要收到客户的货物接收确认证书,导致可能出现商家不发货或提供的货物与所定货物不符的问题和纠纷;另外,SET 协议没有规定在交易结束后,如何安全处理交易数据,当交易出现问题时缺少证据来界定责任等。因此,在实际应用中,完全遵从 SET 协议的电子商务系统很少,往往都针对具体应用环境和安全需求进行了优化或改进,以增强其安全性和适用性。

5.6 公钥基础设施

公钥基础设施(Public Key Infrastructure,PKI)是网络安全建设的基础与重要工作,是电子商务、政务系统安全实施的有力保障。PKI 技术的研究和应用在近几年引起人们的广泛重视,具有得天独厚的发展优势。

PKI 在本质上是建立一种信任环境,通过数字证书来证明某个公开密钥的真实性,并通过证书撤销列表来确认某个公开密钥的有效性。

PKI 技术及其应用还在不断发展中,许多新技术正在不断涌现,证书机构之间的信任模型、使用的加解密算法、密钥管理方案等也在不断变化之中。下面介绍 PKI 应用中的一些问题,包括 PKI 的定义和服务、认证策略、证书管理协议等。

5.6.1 PKI 的定义

PKI 是基于公开密钥理论与技术来实施和提供安全服务的、具有普适性的安全基础设施的总称,它并不特指某一密码设备及其管理设备。一般可认为 PKI 是生成、管理、存储、分发和撤销基于公开密码的公钥证书所需要的硬件、软件、人员、策略和规程的总和。虽然 PKI 的定义在不断延伸和扩展,但一个完整的 PKI 在结构或功能上应该至少包括以下几部分。

1) 证书机构(Certificate Authority,CA)

又称证书管理中心,负责具体的证书颁发和管理,属于可信任的第三方范畴,其作用类

似颁发身份证的机构。CA 可以具有层次结构，除直接管理一些具体的证书之外，还管理一些下级 CA，同时又接受上级 CA 的管理。

CA 作为 PKI 管理实体和服务的提供者，必须经由相关政策来保证其是可信的，并且通过公正的第三方测试从技术上保障其是安全的。CA 通过其数字签名将某个实体的身份信息和相应的公钥捆绑在一起形成公钥证书。CA 是 PKI 框架中唯一能够创建、撤销、维护证书生命期的实体。所有用户都知道用于验证 CA 签名的公钥和用于加密的公钥。

2）注册机构(Registration Authority，RA)

即证书注册审批机构。尽管证书注册的功能可以直接由 CA 来实现，但是用注册机构来实现实体的注册功能是很有意义的。RA 承担 CA 的一定功能并扩展和延伸。它负责证书申请者的信息录入、审核以及证书发放等工作；对发放的证书完成相应的管理功能。发放的数字证书可以存放于智能卡或 U 盘等移动存储介质中。RA 系统是 CA 得以正常运营不可缺少的一部分。

3）证书库

为使用户容易找到所需的公钥证书，必须有一个健壮的、规模可扩充的在线分布式数据库存放 CA 创建的所有用户公钥证书。证书库可由 Web、FTP、X.500 目录等来实现。ISO/ITU、ANSI 和 IETF 等组织制定的标准 X.509 对公钥证书给出定义。X.509 证书适用于大规模网络环境，它的灵活性和可扩展性能够满足各种应用系统不同类型的安全要求。X.509 证书具有支持多种算法、多种命名机制，限制证书的用途，定义证书遵循的策略和控制信任关系的传递等特性。

4）证书撤销列表

在 PKI 环境中公钥证书是有有效期的。在有效期期间如果私钥遭到破坏或用户身份改变，需要有一种方法警告其他用户不能再使用这个公钥。PKI 中引入证书撤销列表(CRL)，该表列出当前已经作废的公钥证书。用户在使用某用户的公钥证书时必须先查 CRL，以便确认公钥的可用性。

5）密钥备份和恢复

在很多环境下(特别是在企业环境下)，由于丢失密钥造成被保护数据的丢失是完全不可接受的。某项业务中的重要文件被对称密钥加密，而对称密钥又被某个用户的公钥加密。假如相应的私钥丢失，导致已加密文件不能解密，则会造成重大损失。一个解决方案就是为多个接收者加密所有数据，但对于高度敏感数据，这个方法是不可行的。一个更可行和通用的可接受方法是备份并能恢复私钥。

6）自动密钥更新

通过用户手工操作的方式定期更新证书在某些应用环境中是不现实的，用户可能忘记自己证书的有效期，只有在认证失败时才发现问题。当用户遇到这些情况时，手工更新过程是极为复杂的，要求与 CA 交换数据(类似于初始化过程)。一种有效解决方法是由 PKI 本身自动完成密钥或证书的更新，完全无须用户干预。无论用户的证书用于何种目的，都会检查有效期，当失效日期到来时，启动更新过程，生成一个新证书来代替旧证书，但用户请求的事务处理继续进行。在很多应用环境下，对可操作的 PKI 来说，自动密钥更新显得十分重要。

7）密钥归档

密钥更新(无论是人工还是自动)意味着经过一段时间，每个用户都会有多个失效的旧

公钥证书和至少一个有效公钥证书。这一系列证书和相应的私钥组成用户的密钥历史档案。记录用户的全部密钥历史档案是很重要的,因为要对多年前加密的文件进行解密(或验证某实体的签名),就需要从历史档案中找出当时使用的解密密钥(或某实体的公钥证书)。在任何系统中,用户自己查找正确的私钥或用每个密钥去尝试解密数据,对用户来说难度很大。PKI必须保存所有密钥,以便正确地备份和恢复密钥,查找出正确的密钥来解密数据。因此密钥归档也应当由PKI自动完成。

8) 交叉认证

建立一个管理全世界所有用户的单一的全球性PKI是事实不可实现的。可能的现实模型是多个PKI独立地运行和操作,为不同的环境和不同的用户团体服务。一个群体中的用户只要有CA的公钥,即可相互验证对方的公钥证书。而不同群体用户间的安全通信就需要一种"交叉认证"机制,即在不同CA间建立信任关系。

9) 支持抗抵赖服务

PKI本身无法提供真正的抗抵赖服务,需要有人工参与分析、判断证据,并做出最后的判决。然而,PKI必须提供所需要的证据、支持决策,提供数据源鉴别和可信的时间戳等信息。

10) 时间戳

支持抗抵赖服务的一个关键条件就是在PKI中使用时间戳。PKI中必须存在用户可信任的权威时间源。事实上,权威时间源提供的时间并不需要精确,仅仅需要用户作为一个参照时间完成基于PKI的事务处理,如事件B发生在事件A的后面。

11) 客户端软件

PKI服务器为用户完成以下工作:CA提供认证服务、资料库保存证书和证书撤销信息、备份和恢复服务器正确管理密钥历史档案、时间戳服务器为文档提供权威时间信息。然而,在客户端/服务器运行模式下,除非客户端发出请求服务。否则服务器通常不会响应客户端。

在用户本地平台上的客户必须请求认证服务。客户端软件必须查询证书和相关的证书撤销信息。客户端软件必须知道密钥历史档案,知道何时需要请求密钥更新或密钥恢复操作。客户端软件必须知道何时为文档请求时间戳。作为安全通信的接收端点,PKI客户端软件需要理解安全策略,需要知道是否、何时和怎样去执行取消操作,需要知道证书路径处理等。客户端软件是全功能、可操作的PKI的必要组成部分。没有客户端软件,PKI无法有效地提供更多服务。而且客户端软件应当独立于所有应用程序之外,完成PKI服务的客户端功能。应用程序通过标准接口与客户端软件连接(与其他基础设施一样),但应用程序本身并不与各种PKI服务器连接,就是说,应用程序使用基础设施,但并不是基础设施的组成部分。

5.6.2 PKI提供的服务和应用

PKI主要提供了以下3种安全服务。

1) 鉴别(认证)服务

PKI提供的鉴别(认证)服务(相对于在本地环境中的非PKI操作的初始化鉴别,包括口令、生物特征扫描设备的单个或多个条件鉴别)采用了数字签名技术,签名产生于以下

3个方面数据的绑定值之上：

(1) 被鉴别的一些数据；

(2) 用户希望发送到远程设备的请求；

(3) 远程设备生成的随机质询信息。

第(1)项支持PKI的数据源鉴别服务，后两项支持PKI的实体鉴别服务。

2) 完整性服务

PKI提供的完整性服务可以采用两种技术。第一种是数字签名，既可以提供实体鉴别，也可以保证被签名数据的完整性。如果签名通过了验证，接收者就认为是收到了原始数据(就是未经修改的数据)。第二种是消息鉴别码(MAC)。这项技术通常采用对称分组密码(如DES-CBC-MAC)或密码散列函数(如HMAC-SHA-1)。

3) 机密性服务

PKI提供的机密性服务采用了类似于完整性服务的机制。例如，要为实体A和实体B之间的通信提供机密性服务，就要：

(1) 实体A随机生成一个对称密钥；

(2) 用对称密钥加密数据；

(3) 将加密后的数据、实体A的公钥以及用实体B的公钥加密后的对称密钥发送给实体B。

PKI提供的上述安全服务恰好能满足电子商务、电子政务、网上银行、网上证券等金融业交易的安全需求，是确保这些活动顺利进行必备的安全措施。没有这些安全服务，电子商务、电子政务、网上银行、网上证券等都无法正常运作。

1. 电子商务应用

电子商务的参与方一般包括买方、卖方、银行和作为中介的电子交易市场。买方通过自己的浏览器上网，登录到电子交易市场的Web服务器并寻找卖方。当买方登录服务器时，互相之间需要验证对方的证书以确认其身份，这被称为双向认证。

在双方身份被互相确认以后，建立起安全通道，并进行讨价还价，之后向商场提交订单。订单里有两种信息：一种是订货信息，包括商品名称和价格；另一种是提交银行的支付信息，包括金额和支付账号。买方对这两种信息进行双重数字签名，分别用商场和银行的证书公钥加密上述信息。当商场收到这些交易信息后，留下订货信息，而将支付信息转发给银行。商场只能用自己专有的私钥解开订货信息并验证签名。同理，银行只能用自己的私钥解开加密的支付信息，验证签名并进行划账。银行在完成划账以后，通知起中介作用的电子交易市场、物流中心和买方，并进行商品配送。整个交易过程都是在PKI所提供的安全服务之下进行，实现了安全、可靠、保密和不可否认性。

2. 电子政务应用

电子政务包含的主要内容有网上信息发布、办公自动化、网上办公、信息资源共享等。按应用模式可分为政府-公众模式(Government to Citizen，G2C)、政府-商务模式(Government to Business，G2B)以及政府-政府模式(Government to Government，G2G)，PKI在其中的应用主要是解决身份认证、数据完整性、数据机密性和抗抵赖性等问题。

例如，一个机密文件发给谁，或者哪一级公务员有权查阅某个机密文件等，这些都需要进行身份认证。与身份认证相关的还有访问控制，即权限控制。认证通过数字证书进行，而

访问控制通过属性证书或访问控制列表(ACL)完成。有些文件在网络传输中要加密以保证数据的机密性,有些文件在网上传输时要求不能丢失和被篡改,特别是一些机密文件的收发必须要有数字签名等。只有PKI提供的安全服务才能满足电子政务中的这些安全需求。

3. 网上银行应用

网上银行是指银行借助互联网技术向客户提供信息服务和金融交易服务。银行通过互联网向客户提供信息查询、对账、网上支付、资金划转、信贷业务、投资理财等服务。网上银行的应用模式有B2C(Business to Customer)个人业务和B2B(Business to Business)对公业务两种。

网上银行的交易方式是点对点的,即客户对银行。客户浏览器端装有客户证书,银行服务器端装有服务器证书。当客户上网访问银行服务器时,银行端首先要验证客户端证书,检查客户的真实身份,确认是否为银行的真实客户;同时服务器还要到CA的目录服务器,通过LDAP协议查询该客户证书的有效期和是否进入"黑名单"。认证通过后,客户端还要验证银行服务器端的证书。双向认证通过以后,建立起安全通道,客户端提交交易信息,经过客户数字签名并加密后传送到银行服务器,由银行后台信息系统进行划账,并将结果进行数字签名返回给客户端。这样就做到了支付信息的机密和完整以及交易双方的不可否认性。

4. 网上证券应用

网上证券广义地讲是证券业的电子商务,包括网上证券信息服务、网上股票交易和网上银证转账等。一般来说,在网上证券应用中,股民为客户端,装有个人证书;券商服务器端装有Web证书。在线交易时,券商服务器只需要认证股民证书,验证是否为合法股民,是单向认证过程。认证通过后,建立起安全通道。股民在网上的交易提交同样要进行数字签名,网上信息要加密传输;券商服务器收到交易请求并解密,进行资金划账并做数字签名,将结果返回给客户端。

5.6.3 PKI的构成

PKI作为分布式计算环境中利用公钥技术所构成的基础安全服务设施,企业或组织可利用相关PKI产品建立安全域,并在其中发布和管理证书。在安全域内,PKI设施管理证书的发布、注册、撤销和更新等。PKI也允许一个组织通过证书级别或直接交叉认证等方式与其他安全域建立信任关系。这些服务和信任关系不局限于独立的网络之内,而建立在网络之间和互联网之上,为电子商务和网络通信提供安全保障。所以具有互操作性的结构化和标准化技术是PKI应用的关键。

PKI在实际应用中是一套软硬件系统和安全策略的集合,它提供了一整套安全机制,使用户在不知道对方身份或分布地很广的情况下,以证书为基础,通过一系列的信任关系进行通信和电子商务交易。

一个典型的PKI系统如图5.5所示,包括PKI安全策略、软硬件系统、证书机构(CA)、注册机构(RA)、密钥与证书管理系统和PKI应用等。

PKI安全策略建立和定义了密码系统使用的处理方法和原则。它包括一个组织怎样处理密钥和机密信息,根据安全风险级别定义安全控制级别。一般情况下,在PKI中有两种类型的策略:一种是证书策略,用于管理证书的使用。例如,可以确认某一CA是互联网上的公有CA,还是某一企业内部的私有CA。另一种是CPS(Certificate Practice Statement)。一

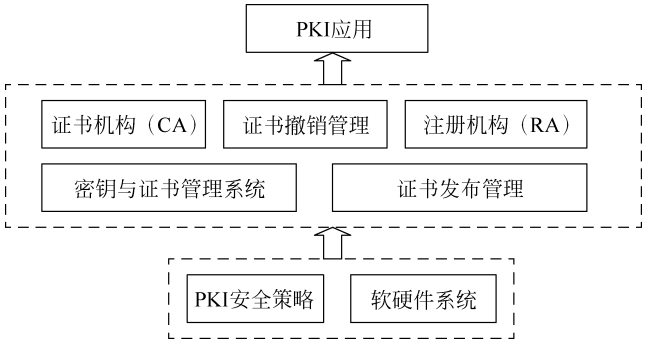


图 5.5　PKI 构成框图

些由商业证书机构(CCA)或者可信第三方操作的 PKI 系统需要 CPS。这是包含如何在实践中增强和支持安全策略的一些操作过程的详细文档。它包括 CA 是如何建立和运行的，证书是如何发布、接收和撤销的，密钥是如何产生、注册的，以及密钥是如何存储的、用户是如何得到密钥的等。

CA 是 PKI 的信任基础，它管理公钥的整个生命周期，其作用包括：发放证书、规定证书的有效期和通过发布证书撤销列表(CRL)确保必要时可以撤销证书。

RA 提供用户和 CA 之间的一个接口，它获取并鉴别用户的身份，向 CA 提出证书请求。它主要完成收集用户信息和确认用户身份的功能。这里的用户是指将要向 CA 申请数字证书的客户，既可以是个人，也可以是集团或团体、某政府机构等。注册管理一般由一个独立的注册机构来承担。它接受用户的注册申请，审查用户的申请资格，并决定是否同意 CA 给其签发数字证书。注册机构并不给用户签发证书，而只是对用户进行资格审查。因此，RA 可以设置在直接面向客户的业务部门，如银行的营业部、机构认证部门等。当然，对于一个规模较小的 PKI 应用系统来说，可将注册管理的职能由 CA 来完成，而不设立独立运行的 RA。但这并不是取消了 PKI 的注册功能，而只是将其作为 CA 的一项功能而已。PKI 国际标准推荐由一个独立的 RA 来完成注册管理的任务，可以增强应用系统的安全。

证书发布系统负责证书的发放。证书发布方法可以通过 LDAP 目录服务来实现。目录服务器可以是一个组织中现存的，也可以是 PKI 方案提供的。

PKI 应用非常广泛，包括 Web 服务器和浏览器之间的通信、电子邮件、电子数据交换(EDI)、互联网上的信用卡交易和虚拟专用网(VPN)等。

一个简单的 PKI 系统包括 CA、RA 和相应的 PKI 存储库。CA 用于签发并管理证书；RA 可以作为 CA 的一部分，也可以独立，其功能包括个人身份审核、CRL 管理、密钥产生和密钥对备份等；PKI 存储库包括 LDAP 目录服务器和普通数据库，用于对用户申请、证书、密钥、CRL 和日志等信息进行存储和管理，并提供一定的查询功能。

一个典型的 CA 系统包括安全服务器、CA 服务器、RA 服务器、LDAP 目录服务器和数据库服务器等，如图 5.6 所示。

安全服务器：面向普通用户，提供证书申请、浏览、证书撤销列表以及证书下载等安全服务。安全服务器与用户的通信采取安全信道方式。用户首先得到安全服务器的证书(该证书由 CA 颁发)，然后用户与服务器之间的所有通信，包括用户填写的申请信息以及浏览器生成的公钥均以安全服务器的密钥进行加密传输。只有安全服务器利用自己的私钥解密

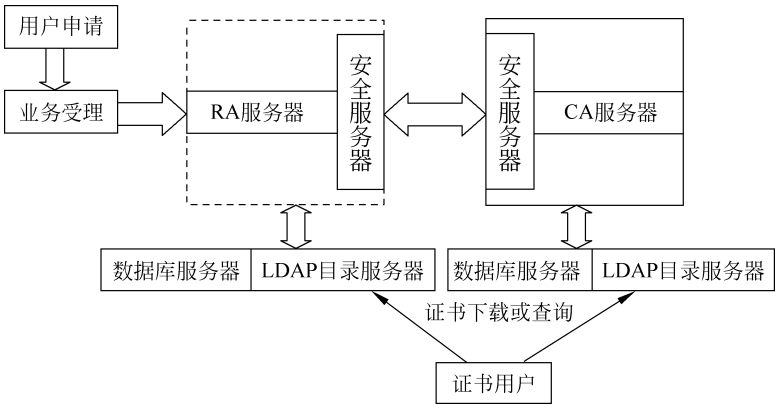


图 5.6 典型 CA 系统框架模型

才能得到明文,这样可以防止其他人通过窃听得到明文,从而保证了证书申请和传输过程中的信息安全。

CA 服务器:整个 CA 的核心,负责证书的签发。CA 服务器首先产生自身的私钥和公钥(密钥长度至少为 1024 位),然后生成数字证书,并且将数字证书传输给安全服务器。CA 还负责为操作员、安全服务器以及 RA 服务器生成数字证书。安全服务器的数字证书和私钥也需要传输给安全服务器。CA 服务器是整个结构中最为重要的部分,存有 CA 的私钥以及发行证书的脚本文件。出于安全的考虑,应将 CA 服务器与其他服务器隔离,任何通信采用人工干预的方式,确保认证中心的安全。

RA 服务器:RA 服务器面向登记中心操作员,在 CA 体系结构中起承上启下的作用。一方面向 CA 服务器转发安全服务器传输过来的证书申请请求,另一方面向 LDAP 目录服务器和安全服务器转发 CA 服务器颁发的数字证书和证书撤销列表。

LDAP 目录服务器:提供目录浏览服务,负责将注册机构服务器传输过来的用户信息以及数字证书加入服务器。这样用户通过访问 LDAP 目录服务器就能够得到其他用户的数字证书。

数据库服务器:CA 中的核心部分,用于 CA 中数据(如密钥和用户信息等)、日志和统计信息的存储和管理。实际的数据库系统应采用多种措施,如磁盘阵列、双机备份和多处理器等,以维护数据库系统的安全性、稳定性、可伸缩性和高性能。

5.6.4 PKI 标准

随着 PKI 技术的不断完善与发展以及应用的日益普及,为了更好地为社会提供优质服务,水平参差不齐的技术供应厂商的 PKI 产品迫切需要解决互联互通问题。此外,PKI 产品自身的安全性也受到越来越多生产厂商和用户的关注。这都要求专门的第三方制定相应的标准规范对 PKI 产品的安全性能进行测评认定。因此,PKI 标准化成为一种必然趋势。

涉及 PKI 的标准可分为两类:一类是与 PKI 定义相关的,另一类是与 PKI 的应用相关的。

1. 与 PKI 定义相关的标准

在 PKI 技术框架中,许多方面都有严格的定义,如用户的注册流程、数字证书的格式、CRL 的格式、证书的申请格式以及数字签名格式等。

国际电信联盟ITU X.509协议是PKI技术体系中应用最广泛、也是最为基础的一个国际标准。它的主要目的是定义一个规范的数字证书格式，以便为基于X.500协议的目录服务提供一种强认证手段。

PKCS(Public-key Cryptography Standards)是由美国RSA数据安全公司及其合作伙伴制定的一组公钥密码学标准。它与ITU X.509协议兼容，也与增强保密邮件(Privacy Enhanced Mail，PEM)协议兼容，并且适应二进制数据的扩充。该标准包括证书申请、证书更新、证书撤销列表(CRL)发布、扩展证书内容以及数字签名、数字信封格式等方面的一系列相关协议。到1999年年底，已经公布了以下PKCS标准：

PKCS#1：RSA公钥函数基本格式标准。它定义了数字签名如何计算，包括待签名数据和签名本身的格式；它也定义了RSA公/私钥的语法。它主要用于PKCS#7中所描述的数字签名和数字信封。

PKCS#3：Diffie-Hellman密钥交换协议。描述了实现Diffie-Hellman密钥协议的方法。

PKCS#5：基于口令的加密标准。描述了使用由口令生成的密钥来加密8位位串并产生一个加密的8位位串的方法。PKCS#5可以用于加密私钥，以便于密钥的安全传输。

PKCS#6：扩展证书语法标准。定义了提供附加实体信息的X.509证书属性扩展的语法格式。

PKCS#7：密码消息语法标准。为使用密码算法的数据规定了通用语法，如数字签名和数字信封。PKCS#7提供了许多格式选项，包括未加密或签名的格式化消息、已封装(加密)消息、已签名消息和既经过签名又经过加密的消息。

PKCS#8：私钥信息语法标准。定义了私钥信息语法和加密私钥语法，其中私钥加密使用了PKCS#5标准。

PKCS#9：可选属性类型。定义了PKCS#6、PKCS#7、PKCS#8和PKCS#10中用到的可选属性类型。已定义的证书属性包括E-mail地址、无格式姓名、内容类型、消息摘要、签名时间、签名副本(Counter Signature)、质询口令字和扩展证书属性。

PKCS#10：证书请求语法标准。定义了证书请求的语法。证书请求包含一个唯一识别名、公钥和可选的一组属性，它们一起被请求证书的实体签名(证书管理协议中的PKIX证书请求消息就是一个PKCS#10)。

PKCS#11：密码令牌接口标准，称为Cryptoki。为拥有密码信息(如加密密钥和证书)和执行密码学函数的单用户设备定义了一个应用程序接口(API)，用于智能卡和PCMCIA卡之类的密码设备。

PKCS#12：个人信息交换语法标准。定义了个人身份信息(包括私钥、证书、各种秘密和扩展字段)的格式。PKCS#12有助于传输证书及对应的私钥，用户可以在不同设备间传输他们的个人身份信息。

PKCS#13：椭圆曲线密码标准。包括椭圆曲线参数的生成和验证、密钥生成和验证、数字签名和公钥加密、密钥协定，以及参数、密钥和方案标识的ASN.1语法。

PKCS#14：伪随机数生成标准。PKI中用到的许多基本的密码学函数，如密钥生成和Diffie-Hellman共享密钥协商，都需要使用随机数。如果“随机数”不是随机的，而是取自一个可预测的取值集合，那么密码学函数的安全性将受到极大影响。因此，安全伪随机数的生

成对于 PKI 的安全极为关键。

PKCS#15：密码令牌信息格式标准。PKCS#15 通过定义令牌上存储的密码对象的通用格式来增进密码令牌的互操作性。

PKIX(Public Key Infrastructure on X.509)是由 IETF 组织中的 PKI 工作小组制定的系列国际标准。它定义了 X.509 证书在互联网上的使用，证书的生成、发布和获取，各种产生和分发密钥的机制，以及怎样实现这些标准的轮廓结构等。

OCSP(Online Certificate Status Protocol)是 IETF 颁布的用于检查数字证书在某一交易时刻是否仍然有效的标准。该标准提供给 PKI 用户一条方便快捷的数字证书状态查询通道，使 PKI 体系能够更有效、更安全地在各个领域中被广泛应用。

LDAP 轻量级目录访问协议简化了笨重的 X.500 目录访问协议，并且在功能性、数据表示、编码和传输方面都进行了相应的修改。1997 年，LDAP v.3 成为互联网标准。目前，LDAP v.3 已经在 PKI 体系中被广泛用于证书信息发布、CRL 信息发布、CA 政策以及与信息发布相关的其他方面。

2001 年，Microsoft、Versign 和 webMethods 3 家公司共同发布了 XML 密钥管理规范 XKMS，被称为第二代 PKI 标准。XKMS 由两部分组成：XML 密钥信息服务规范 X-KISS 和 XML 密钥注册服务规范 X-KRSS。XKMS 通过向 PKI 提供 XML 接口使用户从烦琐的配置中解脱出来，开创了一种新的信任服务。目前，XKMS 已经成为 W3C 的推荐标准，并被 Microsoft、Versign 等公司集成于所属的产品中。

2. 与 PKI 应用相关的标准

除了与 PKI 定义相关的标准外，还有一些构建在 PKI 体系上的应用标准或协议。国际上相继发布了多个与 PKI 应用领域相关的标准，如安全的套接层协议(SSL)、运输层安全协议(TLS)、安全的多用途互联网邮件扩展协议(S/MIME)和 IP 安全协议(IPSec)等。

S/MIME 是一个用于发送安全报文的 IETF 标准。它采用了 PKI 数字签名技术并支持消息和附件的加密，无须收发双方共享相同密钥。S/MIME 委员会采用 PKI 技术标准来实现 S/MIME，并适当扩展了 PKI 功能。目前该标准包括密码报文语法、报文规范、证书处理以及证书申请语法等方面的内容。

SSL/TLS 是互联网中访问 Web 服务器最重要的安全协议。当然，它们也可以应用于基于客户机/服务器模型的非 Web 类型的应用系统。SSL/TLS 都利用 PKI 的数字证书来认证客户和服务器的身份。

IPSec 是 IETF 制定的 IP 层加密协议，PKI 技术为其提供了加密和认证过程的密钥管理功能。IPSec 协议常常用于开发安全 VPN。

目前，PKI 体系中已经包含了众多的标准和协议。由于 PKI 技术的不断进步和完善，以及其应用的不断普及，将来还会有更多的标准和协议加入。

5.6.5 PKI 的信任模型

选择信任模型(Trust Model)是构筑和运作 PKI 所必需的一个环节。选择正确的信任模型以及与它相应的安全级别是非常重要的，同时也是部署 PKI 所要做的较早和基本的决策之一。信任模型主要阐述了以下几个问题：

(1) 一个 PKI 用户能够信任的证书是怎样被确定的？

(2) 这种信任是怎样被建立的?

(3) 在一定的环境下,这种信任如何被控制?

在详细介绍 PKI 信任模型之前,先介绍几个相关的概念。

信任:在 ITU-T 推荐标准 X.509 规范中,对"信任"的定义是:如果实体 A 认为实体 B 会严格地如它期望的那样行动,则实体 A 信任实体 B。其中的实体是指在网络或分布式环境中具有独立决策和行动能力的终端、服务器或智能代理等。在 PKI 中,可以把这个定义具体化为:如果一个用户假定 CA 可以把任一公钥绑定到某个实体上,则他信任该 CA。

信任域:信任域是指一个组织内的个体在一组公共安全策略控制下所能信任的个体集合。个体可以是个人、服务器以及具体的应用程序等。公共安全策略是指系统颁发、管理和验证证书所依据的一系列规定、规则的集合。

信任模型:信任模型是指建立信任关系和验证证书时寻找和遍历信任路径的模型。信任关系并不总是直接建立在域中两个实体之间,而是建立在一定长度的信任路径之上。信任路径是指由域中多个相邻的值得信赖的个体组成的链路。

信任锚:在信任模型中,当可以确定一个实体的身份或者有一个足够可信的身份签发者证明该实体的身份时,才能作出信任该实体身份的决定。这个可信的身份签发者称为信任锚。简单地说,信任锚就是 PKI 信任模型中的信任起点。

1. 单 CA 信任模型

这是最基本的信任模型,也是企业环境中比较实用的一种模型。如图 5.7 所示,在这种模型中,整个 PKI 体系只有一个 CA,它为 PKI 中的所有终端实体签发和管理证书。PKI 中的所有终端实体都信任这个 CA。每个证书路径都起始于该 CA 的公钥,该 CA 的公钥成为 PKI 体系中唯一的用户信任锚。

单 CA 信任模型的主要优点是:容易实现,易于管理,只需要建立一个根 CA,所有的终端实体都能实现相互认证。

这种信任模型的局限性也十分明显:不易扩展支持大量的或者不同的群体用户。终端实体的群体越大,支持所有必要的应用就越困难。

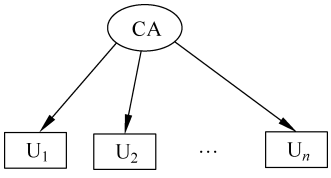


图 5.7　单 CA 信任模型

2. CA 的严格层次结构信任模型

如图 5.8 所示,CA 的严格层次结构可以描绘为一棵倒转的树,根在顶上,树枝向下伸展,树叶在下面。在这棵倒转的树上,根代表一个对整个 PKI 域内的所有实体都有特别意义的 CA——通常被叫做根 CA。作为信任的根或信任锚,所有实体都信任它。根 CA 通常不直接为终端实体颁发证书,而只为子 CA 颁发证书。

在根 CA 的下面是零层或多层子 CA,子 CA 是所在实体集合的根。与非 CA 的 PKI 实体相对应的树叶即为终端实体。两个不同的终端实体进行交互时,双方都提供自己的证书和数字签名,通过根 CA 对证书进行有效性和真实性的认证。在 CA 的严格层次结构中,信任关系是单向的,上级 CA 可以而且必须认证下级 CA,而下级 CA 不能认证上级 CA。显然,单 CA 信任模型是严格层次结构的一种特例。

严格层次结构信任模型中的每个实体(包括子 CA 和终端实体)都必须安全拥有根 CA 的公钥。下面通过一个例子说明在该信任模型中进行认证的过程。

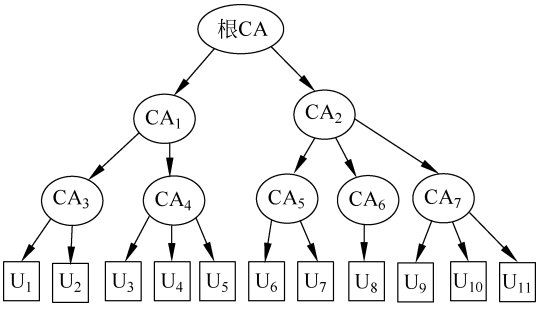


图 5.8 CA 的严格层次结构信任模型

一个持有根 CA 公钥 PK_R 的终端实体 A 可以通过下述方法验证另一个终端实体 B 的证书。假设实体 B 的证书是由 CA_2 签发的,而 CA_2 的证书是由 CA_1 签发的,CA_1 的证书是由根 CA 签发的。实体 A 利用根 CA 公钥 PK_R 能够验证 CA_1 的公钥 PK_1,因此可以从 CA_1 的证书获取到 CA_1 的可信公钥。然后,这个公钥又可以被实体 A 用来验证 CA_2 的公钥。类似地,就可以获取到 CA_2 的可信公钥 PK_2。这样,公钥 PK_2 就能够被实体 A 用来验证实体 B 的证书,从而得到实体 B 的可信公钥 PK_B。实体 A 现在就可以根据密钥的类型来使用密钥 PK_B。例如,对发给实体 B 的消息进行加密,或者验证据称是由实体 B 所签署的数字签名,从而实现实体 A 和实体 B 之间的安全通信。

CA 的严格层次结构信任模型由于其简单的结构和单向的信任关系,具有以下优点:容易增加新的信任域用户;证书路径具有单向性,到达一个特定最终实体只有唯一的信任路径;证书路径相对较短;同一机构中信任域扩展容易。

这种信任模型的主要局限性是:单个 CA 的失败会影响整个 PKI 体系,而且与根 CA 的距离越小,造成的混乱越大;由于所有的信任都集中在根 CA,一旦该 CA 出现故障(如私钥泄露),整个信任体系就会瓦解。而且对于开放式网络而言,建造一个统一的根 CA 也是不现实的。

3. CA 的分布式信任模型

分布式信任模型也叫网状信任模型,在这种模型中,CA 间存在着交叉认证。如果任何两个 CA 间都存在着交叉认证,则这种模型就成为严格网状信任模型。

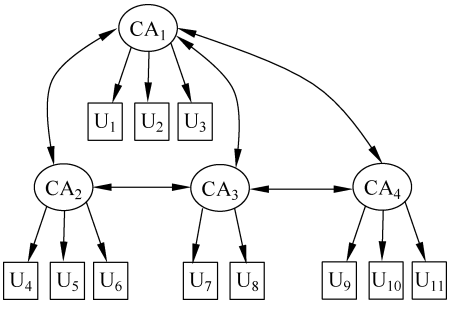


图 5.9 CA 的分布式信任模型

与在 PKI 体系中所有实体都信任唯一根 CA 的严格层次结构信任模型不同,分布式信任模型中没有一个所有用户都唯一信任的根 CA,而是把信任分散到若干个 CA 上。如图 5.9 所示,每个实体都将自己的发证 CA 作为信任的起点,各 CA 之间通过颁发交叉证书进行交叉认证,从而扩展信任关系,最终形成一种网状结构。在该模型中,每个 CA 直接对其所属用户实体颁发证书。CA 之间没有上下属关系,而是通过交叉证书联系在一起。因此,如果没有命名空间的限制,那么任何 CA 都可以对其他 CA 发证。当一个组织想要整合各个独立开发的 PKI 体系时,采用这种模型结构较适合。

分布式信任模型的最大特点在于具有更好的灵活性,任何一个独立 CA,只要与网内的

某一 CA 建立了交叉认证，即可加入，因而信任域的扩展性也较好；此外，由于存在多个信任锚，单个 CA 安全性的削弱不会影响整个 PKI，因此信任建立的安全性较高。另外，增加新的认证域较为容易，新的根 CA 只须在网中至少向一个 CA 发放过证书，用户不需要改变信任锚。

但也正是由于分布式信任模型的这种灵活性，使整个系统的可管理性变差。随着 CA 数量的增加，证书路径可能较长，信任的传递将引起信任度的衰减，信任关系处理的复杂度增加；还可能出现多条证书路径和死循环的现象，这使得证书验证变得困难。

4. CA 的 Web 信任模型

这种模型建构在浏览器（如微软公司的 Internet Explorer）的基础上，浏览器厂商在浏览器中内置了多个根 CA，每个根 CA 相互间是平行的，浏览器用户信任这多个根 CA，并把这多个根 CA 作为自己的信任锚。

如图 5.10 所示，这种模型看上去与分布式信任模型颇为相似，实际上，它更接近 CA 的严格层次结构模型。它通过与相关域进行互连而不是扩大现有的主体群，使客户实体成为浏览器中所给出的所有域的依托方。各个嵌入的根 CA 并不被浏览器厂商显式认证，而是物理地嵌入软件来发布，作为对 CA 名字和它的密钥的安全绑定。但是各个根 CA 是浏览器厂商内置的，浏览器厂商隐含认证了这些根 CA。这样，浏览器厂商就成为事实上隐含的根 CA。

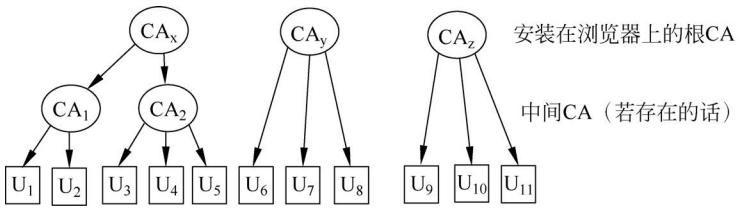


图 5.10　CA 的 Web 信任模型

CA 的 Web 信任模型的优点主要有：方便简单，操作性强，对终端用户的要求较低，用户只需简单地信任嵌入的各个根 CA。

这种模型的主要缺点如下。

（1）安全性较差。如果这些根 CA 中有一个是坏的，即使其他根 CA 完好无损，安全性也将被破坏。

（2）根 CA 与终端用户信任关系模糊。终端用户与嵌入的根 CA 间交互十分困难。

（3）扩展性差。根 CA 都已预先安装，难以扩展。

（4）让用户来管理如此多的公钥增加了用户负担，对用户的技术水平要求较高。

5. 以用户为中心的信任模型

如图 5.11 所示，在以用户为中心的信任模型中，每个用户都直接决定信任哪个证书和拒绝哪个证书。没有可信的第三方作为 CA，终端用户就是自己的根 CA。

下面用 PGP 来说明以用户为中心的信任模型。在 PGP 中，一个用户通过担当 CA（签发其他实体的公钥）和使其公钥被其他人所认证来建立所谓的信任网。例如，当用户 U_2 收到一个号称属于用户 U_7 的公钥证书时，发现这个证书是由不认识的用户 U_6 签发的，用户 U_6 的证书又是由认识并信任的用户 U_5 签发的（用户 U_5 有由用户 U_2 签发的公钥证书）。

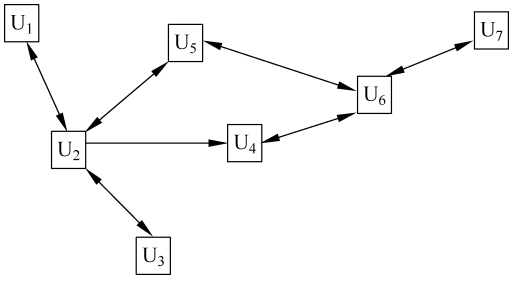


图 5.11 以用户为中心的信任模型

通过从用户 U_5 到用户 U_6 再到用户 U_7 的密钥链信任,用户 U_2 就可以决定信任用户 U_7 的密钥,当然也可以决定不接受用户 U_7 的公钥。

以用户为中心的信任模型的优点如下:

(1) 安全性很强。在高技术性和高利害关系的群体中,这种模型具有优势。

(2) 用户可控性很强。用户可自己决定是否信赖某个证书。也就是说,每个用户对决定信赖哪个证书和拒绝哪个证书直接完全负责。

这种模型的主要局限性是使用范围较窄。对于没有多少安全知识或 PKI 概念的一般群体,将发放和管理证书的任务交给用户是不现实的,不适合采用以用户为中心的信任模型。另外,这种模型也不适合有严格组织机构的群体。因为在这些群体中,往往需要以集约的方式控制一些公钥,而不希望完全由用户自己控制。这样的组织信任策略在以用户为中心的信任模型中不能用一种自动的和可实施的方式来实现。

关于 PKI 的信任模型问题,需要注意的是:信任模型实质上是决定实体对信任锚的选择。在 CA 的严格层次结构信任模型中,实体的信任锚是逻辑上离它最远的 CA(在层次结构根上的 CA);在分布式信任模型中,实体的信任锚是逻辑上离它最近的 CA;在 Web 信任模型中,实体的信任锚实际是一组锚(CA 的根密钥预装在浏览器中);在以用户为中心的信任模型中,实体的信任锚则是自己选择的一个或多个 CA。

信任模型的选择对于能否全面信任所赋予的目标证书公钥起着重要的作用,因此选择合理的信任模型和确定适当的模型结构是建设 PKI 平台的重要问题。

5.7 秘密分割

在密码学的应用领域存在这样一类问题:一个重要消息被加密后,解密这个重要消息需要两个(或两个以上)拥有密钥的实体共同协作才能完成,任何一个人都无法单独对加密的信息进行解密。为了解决这一问题,人们提出一种称为“秘密共享”的解决方案。

现实生活中存在大量这类问题。例如,某银行有 1 位正行长和 5 位副行长。如果每位正副行长都有可以打开银行金库的钥匙,虽然很方便但却不安全,容易导致安全事件。如果所有或指定的银行领导到场并使用各自的钥匙才能打开金库,虽然安全,但却很不方便,如可能因为某位领导出差而无法打开银行金库。一种折中的解决办法是给每位领导一个钥匙,但至少有 3 位领导同时到场并使用各自的钥匙才能打开银行金库。

当然,安全控制策略还可以更灵活一些。例如,可以给正行长两把钥匙,副行长一把钥匙;要求正行长和任意副行长一起就能打开金库;或者任意 3 位副行长一起也能打开金库;

而只有正行长，或只有3位以下的副行长却不能打开金库。

为了解决这类问题，人们从数学上提出了一种称为“秘密分割”的解决方案，也称为阈值方案、门限方案(Threshold Scheme)。

所谓秘密分割方案是指将一个秘密 s 分成 n 个子密钥 $s_1, s_2, \cdots, s_n$，也称为影子(Shadow)或份额(Share)，并安全分配给 n 个参与者分别持有，使得满足下列两个条件，则称这种方案为(k,n)门限方案，其中 k 为门限值。

(1) 由任意 k 个或多于 k 个参与者拥有的 s_i 可重构秘密 s。

(2) 由任意 $k-1$ 个或少于 $k-1$ 个参与者拥有的 s_i 不可重构秘密 s。

如果一个参与者或一组未经授权的参与者在猜测秘密 s 时，并不比局外人猜测秘密 s 时有优势，即满足条件(3)：少于 k 个参与者所持有的部分信息得不到秘密 s 的任何信息，则称这个方案是完善的。

由于重构密钥至少需要 k 个子密钥，故暴露 $r(r \leqslant k-1)$个份额也不会危及秘密。少于 k 个参与者的共谋也不能得到密钥。另外，若某个份额偶然丢失或破坏，仍可以恢复密钥(只要至少有 k 个有效份额)。秘密分割方案在密钥分配中是有典型意义的。

从信息的角度看，$\{m_1, m_2, \cdots, m_n\}$是$(k,n)$门限方案等价于：

1) 对$\{m_1, m_2, \cdots, m_n\}$中任意 $u \geqslant k$ 个 $m_{i1}, m_{i2}, \cdots, m_{iu}$，均有 $H(s \mid m_{i1}, m_{i2}, \cdots, m_{iu}) = 0$。即知道 $m_{i1}, m_{i2}, \cdots, m_{iu}$ 时，s 的不确定性为0，即 s 被完全确定。

2) 对$\{m_1, m_2, \cdots, m_n\}$中任意 $v < k$ 个 $m_{i1}, m_{i2}, \cdots, m_{iv}$，均有 $H(s \mid m_{i1}, m_{i2}, \cdots, m_{iv}) = H(s)$。即知道 $m_{i1}, m_{i2}, \cdots, m_{iv}$ 时，s 的不确定性同什么都不知道时完全一样。

下面介绍最具代表性的两个秘密分割门限方案。

5.7.1　Shamir秘密分割门限方案

Shamir秘密分割门限方案基于多项式的拉格朗日(Lagrange)插值公式，其基本思路如下：设$\{(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\}$是平面上 k 个点构成的集合，其中 $x_i (i = 1, 2, \cdots, k)$均不相同，那么在平面上存在一个唯一的 $k-1$ 次多项式 $f(x)$ 通过这 k 个点，即 $f(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$。

若把秘密 s 取作 $f(0)$，即 a_0，n 个子密钥取作 $f(x_i)(i = 1, 2, \cdots, n)$，那么利用其中的任意 k 个子密钥可重构 $f(x)$，从而得到密钥 s(即 a_0)。

Shamir秘密分割门限方案由系统初始化、秘密分发和秘密重构3个阶段组成。

1. 系统初始化

设 GF(q)是一有限域，其中 q 是一大素数，满足 $q \geqslant n+1$(n 为系统参与者数)。秘密 s 是在 GF(q)\{0}上均匀选取的一个随机数，表示为 $s \in_R$ GF(q)\{0}。

在 GF(q)上构造一个 $k-1$ 次多项式：$f(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$。其中，$k-1$ 个随机选取的系数 $a_1, a_2, \cdots, a_{k-1}$ 也满足 $a_i \in_R$ GF(q)\{0}$(i = 1, 2, \cdots, k-1)$。将 q 作为系统参数公开，而多项式 $f(x)$保密。设 n 个参与者为 $P_1, P_2, \cdots, P_n$，身份标识为 $i_1, i_2, \cdots, i_n$，且互不相同。

2. 秘密分发

秘密分发者利用各个用户标识 $i_k (1 \leqslant k \leqslant n)$和 $f(x)$进行计算。将计算结果 $f(i_k)(1 \leqslant k \leqslant n)$分发给各个用户，作为用户子密钥。

3. 秘密重构

k 个参与者利用各自的“子密钥”$f(i_k)$和身份标识 $i_k(1\leqslant k\leqslant n)$构造如下线性方程组：

$$\begin{cases} a_0+a_1(i_1)+\cdots+a_{k-1}(i_1)^{k-1}=f(i_1) \\ a_0+a_1(i_2)+\cdots+a_{k-1}(i_2)^{k-1}=f(i_2) \\ \vdots \\ a_0+a_1(i_k)+\cdots+a_{k-1}(i_k)^{k-1}=f(i_k) \end{cases}$$

写成矩阵的形式：

$$\begin{bmatrix} 1 & i_1 & \cdots & i_1^{k-1} \\ 1 & i_2 & \cdots & i_2^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & i_k & \cdots & i_k^{k-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{bmatrix} = \begin{bmatrix} f(i_1) \\ f(i_2) \\ \vdots \\ f(i_k) \end{bmatrix}$$

设系数矩阵为 $\boldsymbol{A}$，显然，$\boldsymbol{A}$ 是一个 Vandermonde 矩阵。因为 $i_k(1\leqslant k\leqslant n)$均不相同，$\det(\boldsymbol{A})\neq 0$，因此该线性方程组有唯一解。

可由拉格朗日插值公式构造如下多项式：

$$f(x)=\sum_{j=1}^{k} f(i_j)\prod_{\substack{l=1\\ l\neq j}}^{k}\frac{(x-i_l)}{(i_j-i_l)} \bmod q$$

显然，只要知道 $f(x)$，易于计算出秘密 $s=f(0)$。其中，加、减、乘、除运算都是在 $\mathbf{Z}_P$ 上运算，除法是乘以分母的逆元素。

$$\begin{aligned} f(i_1)&=a_{k-1}(i_1)^{k-1}+a_{k-2}(i_1)^{k-2}+\cdots+a_1(i_1)+a_0 \\ f(i_2)&=a_{k-1}(i_2)^{k-1}+a_{k-2}(i_2)^{k-2}+\cdots+a_1(i_2)+a_0 \\ &\vdots \\ f(i_n)&=a_{k-1}(i_n)^{k-1}+a_{k-2}(i_n)^{k-2}+\cdots+a_1(i_n)+a_0 \end{aligned}$$

实际上参与者仅需知道 $f(x)$的常数项 $f(0)$，而无须知道整个多项式 $f(x)$，故仅需以下表达式就可重构出秘密 s：

$$s=(-1)^{k-1}\sum_{j=1}^{k} f(i_j)\prod_{\substack{l=1\\ l\neq j}}^{k}\frac{i_l}{(i_j-i_l)} \bmod q$$

4. 安全性分析

如果 $k-1$ 个参与者想获得秘密 s，他们可构造出由 $k-1$ 个方程构成的线性方程组，其中有 k 个未知量。

对 GF(q)中的任一值 s_0，可设 $f(0)=s_0$，这样可得到第 k 个方程，并由拉格朗日插值公式得出 $f(x)$。因此对每一 $s_0\in \mathrm{GF}(q)$都有一个唯一的多项式满足线性方程组，所以已知 $k-1$ 个子密钥得不到关于秘密 s 的任何信息，因此这个方案是完善的。

Shamir 秘密分割门限方案的主要特点如下：

(1) 它是完善的门限方案；

(2) 每个份额的大小与秘密值的大小相近；

(3) 易于扩充新用户，即计算要分配的新份额不影响原来的各个份额；

(4) 安全性不依赖于未经证明的假设；

(5) 门限值固定；

(6) 秘密分发者知道参与者的份额；

(7) 不能防止秘密分发者和参与者的欺诈。

【例 5.2】 设有 5 个参与者，标识为 1、2、3、4、5，实现(3,5)秘密分割门限方案。

这里，$k=3$，$n=5$。假设系统选择参数 $q=19$，秘密 $s=11$。随机选取 $a_1=2$，$a_2=7$，得秘密多项式为

$$f(x)=(7x^2+2x+11)\bmod 19$$

分发者分别进行如下计算：

$$f(1)=(7+2+11)\bmod 19=20\bmod 19=1$$
$$f(2)=(28+4+11)\bmod 19=43\bmod 19=5$$
$$f(3)=(63+6+11)\bmod 19=80\bmod 19=4$$
$$f(4)=(112+8+11)\bmod 19=131\bmod 19=17$$
$$f(5)=(175+10+11)\bmod 19=196\bmod 19=6$$

分发者将 $f(1)$、$f(2)$、$f(3)$、$f(4)$、$f(5)$分发给各个用户。如果知道其中的 3 个子密钥，如 $f(1)=1$，$f(2)=5$，$f(3)=4$，就可按以下方式重构 $f(x)$：

$$1\frac{(x-2)(x-3)}{(1-2)(1-3)}=\frac{(x-2)(x-3)}{2}=(2^{-1}\bmod 19)(x-2)(x-3)$$

$$5\frac{(x-1)(x-3)}{(2-1)(2-3)}=5\frac{(x-1)(x-3)}{-1}=5((-1)^{-1}\bmod 19)(x-1)(x-3)$$

$$4\frac{(x-1)(x-2)}{(3-1)(3-2)}=4\frac{(x-1)(x-2)}{2}=4(2^{-1}\bmod 19)(x-1)(x-2)$$

可计算得 $2^{-1}\equiv 10 \bmod 19$，$(-1)^{-1}\equiv 18 \bmod 19$。

所以 $f(x)=[10(x-2)(x-3)+90(x-1)(x-3)+40(x-1)(x-2)]\bmod 19$

$$=[10(x^2-5x+6)+14(x^2-4x+3)+2(x^2-3x+2)]\bmod 19$$
$$=7x^2+2x+11$$

可得秘密信息为 $s=f(0)=11$。

5.7.2　Asmuth-Bloom 门限方案

在密钥分配方案中，出现不少新的改进方案。1983 年，Asmuth-Bloom 基于中国剩余定理提出了一个门限方案。其中的子密钥是与秘密数据(如共享密钥)相关联的一个数的同余类。该(k,n)门限方案描述如下。

设秘密数据为一正整数 s，选取一大素数 $q(q>s)$以及 n 个严格递增的正整数 $m_1,m_2,\cdots,m_n$，满足：

(1) $(m_i,m_j)=1$　$(\forall i,j,i\neq j)$。

(2) $(q,m_i)=1(\forall i)$，即 m_i 不能是 q 的整倍数。

(3) $N=\prod_{i=1}^{k}m_i>q\prod_{i=1}^{k-1}m_{n-i+1}$，其中 N 为 k 个最小的 m_i 之积。显然，由于$\{m_i\}$为严

格递增序列,所以任意 k 个 m_i 的乘积 N' 一定大于 N。

进行秘密分发时,随机选取整数 A,满足 $0\leqslant A\leqslant[N/q]-1$,公布大素数 q 和整数 A。

(4) 分发者先计算 $y=s+Aq$,$y_i\equiv y \bmod m_i(i=1,2,\cdots,n)$,$(m_i,y_i)$ 即为第 i 个子密钥(影子),集合 $\{(m_i,y_i)\}_{i=1}^{n}$ 即构成了一个 (k,n) 门限方案。

显然,为了恢复 s,只要找到 y 就足够了。

下面验证方案的正确性。

当 k 个参与者(记为 $i_1,i_2,\cdots,i_k$)提供出自己的子密钥时,由 $\{(m_{i_j},y_{i_j})\}_{j=1}^{k}$ 建立方程组:

$$\begin{cases} x\equiv y_{i_1} \bmod m_{i_1} \\ x\equiv y_{i_2} \bmod m_{i_2} \\ \vdots \\ x\equiv y_{i_k} \bmod m_{i_k} \end{cases}$$

根据中国剩余定理可求得 $x\equiv y' \bmod N'$,式中 $N'=\prod_{j=1}^{k} m_{i_j}\geqslant N$。

由中国剩余定理知,y' 是模 N' 下同余方程的唯一解。

这时,注意秘密分发时所随机选取的整数 A 满足 $0\leqslant A\leqslant[N/q]-1$,即 $Aq<N-q$,因此 $y=s+Aq<s+N-q$。又由 $q>s$,即 $s-q<0$,得 $y\leqslant s+N-q<N$,而 $N'\geqslant N$,可得 $y<N'$。

因此由中国剩余定理解出的 $y'=y$ 是唯一的,再计算 $s=y-Aq$ 即可恢复出秘密数据 s。

但是,若仅有 $k-1$ 个参与者提供出自己的子密钥 (m_i,y_i),由 $\{(m_{i_j},y_{i_j})\}_{j=1}^{k-1}$ 建立方程组,则只能求得 $x\equiv y''(\bmod N'')$,式中 $N''=\prod_{j=1}^{k-1} m_{i_j}$。这时,$y''$ 是仅由 $k-1$ 个子密钥确定的同余方程的解。

由于 $\{m_i\}$ 为严格递增序列,故 $N''=\prod_{j=1}^{k-1} m_{i_j}<\prod_{i=1}^{k-1} m_{n-i+1}$。由条件(3),$N=\prod_{i=1}^{k} m_i>q\prod_{i=1}^{k-1} m_{n-i+1}$,有 $\frac{N}{q}>\prod_{i=1}^{k-1} m_{n-i+1}>\prod_{j=1}^{k-1} m_{i_j}=N''$,因此 $N''<\frac{N}{q}$ 或 $\frac{N}{N''}>q$。

由于 m_i 不能是素数 q 的整倍数,所以 $(N'',q)=1$,故集合 $\{[x]\mid[x]=\{x:x \bmod N''\equiv y''\}\}$ 将覆盖所有的模 q 同余类。或者更简单地说,至少以下整数 $y''+\alpha N''(0\leqslant\alpha<q)$ 都是 y 的可能取值,因此无法准确恢复 y。

【例 5.3】 设 $k=2,n=3,q=7,s=4,m_1=9,m_2=11,m_3=13$,则 $N=m_1\ m_2=99>q\times m_3=7\times13=91$。

在 $[0,[99/7]-1]=[0,13]$ 之间随机地取 $A=8$,求 $y=s+Aq=4+8\times7=60$,$y_1\equiv y \bmod m_1\equiv60 \bmod 9\equiv6$,$y_2\equiv y \bmod m_2\equiv60 \bmod 11\equiv5$,$y_3\equiv y \bmod m_3\equiv60 \bmod 13\equiv8$。$\{(9,6),(11,5),(13,8)\}$ 即构成(2,3)门限方案。

若知道 $\{(9,6),(11,5)\}$,可建立方程组

$$\begin{cases} y\equiv 6 \bmod 9 \\ y\equiv 5 \bmod 11 \end{cases}$$

利用中国剩余定理，可解得 $y\equiv(M_1' M_1 b_1+M_2' M_2 b_2)\bmod M\equiv(11\times5\times6+9\times5\times5)\bmod 99\equiv60$。所以，秘密信息 $s=y-Aq=60-8\times7=4$。

5.8 群　密　钥

群密钥(Group Key)又称为组密钥或会议密钥(Conference Key)，是用于提供多方共享的密钥。在互联网上各种协作式应用中，如多媒体电子会议、游戏和数据库等，一个多方共享的密钥会将数据加密、数据完整性和实体鉴别等服务变得简单有效。如何有效地建立和管理这种多方共享密钥是安全服务的关键。

建立群密钥的困难主要在于以下几个问题：

- 群组成员的加入和离开；
- 群组的合并和分离；
- 叠加问题(在一个成员加入时又有其他成员加入)；
- 通信和计算时间代价；
- 周期地重新建立共享密钥。

下面介绍一种典型的群密钥协议。该协议中有 n 个用户，允许任意 t 个用户组成的群组通过不安全信道推导出一个共享的群密钥。这个群密钥是动态的，且不同的 t 个用户生成的群密钥不同。

假定 n 个用户中的任意 t 个成员的标识符为 $U_0, U_1, \cdots, U_{t-1}$，$p$、$q$ 是用户共享的两个大素数，其中 $q|(p-1)$，而 α 是 $\mathbf{Z}_p^*$ 的 q 阶元。

群密钥生成协议如下：

(1) 每个用户 $U_i(i=0,1,2,\cdots,t-1)$ 选取一个随机数 $r_i\in\{1,2,\cdots,q-1\}$，计算 $z_i=\alpha^{r_i}\bmod p$，并将 z_i 广播分发给该组其他 $t-1$ 个成员；

(2) 每个用户 U_i 验证 $z_i^q\equiv1 \bmod p$；

(3) 每个用户 U_i 接收到用户 U_{i-1}、U_{i+1} 的消息 z_{i-1} 和 z_{i+1}，计算 $x_i=(z_{i+1}/z_{i-1})^{r_i}\bmod p$，并将 x_i 分发给该组其他 $t-1$ 个成员；

(4) 每个用户 U_i 接收到 $x_j(0\leqslant j\neq i\leqslant t-1)$，计算

$$K=K_i=(z_{i-1})^{t\cdot r_i}x_i^{t-1}x_{i+1}^{t-2}\cdots x_{i+(t-2)}^{t-2}\bmod p$$

在协议的步骤(2)和步骤(3)中，下标 $i-1$、$i-2$ 和 $i+1$ 等的计算都是关于模 t 的运算。

可以证明，当协议执行完后，群组内所有合法用户都可以计算出相同的群密钥 $K=\alpha^{r_0r_1+r_1r_2+\cdots+r_{t-2}r_{t-1}+r_{t-1}r_0}\bmod p$，而群组外的其他人不能获取任何有用的信息。

特别要指出的是，当 $t=2$ 时，协议生成的群密钥为 $K=(\alpha^{r_0r_1})^2\bmod p$，恰好是标准 Diffie-Hellman 密钥交换协议密钥的平方。

该协议的主要局限是无法抵抗中间人攻击。针对这个问题，近年来出现了不少好的群密钥分发协议，读者可以参阅其他有关文献。

思考题与习题

5-1 为什么要进行密钥管理?

5-2 为什么在密钥管理中要引入层次式结构?

5-3 密钥管理的原则是什么?

5-4 密钥分配的安全性应当注意什么问题?公开密钥分发与秘密密钥分发的区别是什么?

5-5 密钥分发与密钥协商过程的目标是什么?各有何特点?

5-6 简述 Diffie-Hellman 密钥协商协议的原理和局限。

5-7 设 Diffie-Hellman 密钥交换协议的公开参数 $p=719$ 和 $\alpha=3$,用户 A 和用户 B 想建立一个共享的秘密密钥 K_s。若用户 A 向用户 B 发送 191,用户 B 以 543 回答。设用户 A 的秘密随机数 $r_A=16$,求建立的秘密密钥 K_s。

5-8 在 Diffie-Hellman 密钥交换过程中,设大素数 $p=11$,$\alpha=2$ 是 Z_p 的本原元。用户 U 选择的随机数是 5,用户 V 选择的随机数是 7,试确定 U 和 V 之间的共享密钥。

5-9 什么是公钥数字证书,它主要包括哪些基本内容?在应用时该如何确认证书及证书持有人的有效性和真实性?

5-10 什么是 PKI?简述 PKI 技术的主要意义。

5-11 简述 PKI 提供的服务功能和系统组成。

5-12 什么是 PKI 的信任模型?有哪几种典型的信任模型?各有何特点?

5-13 在 Shamir 门限方案中,设 $p=17$,$k=3$,$n=5$,秘密 $s=13$,选取 $a_1=2$,$a_2=7$,建立秘密多项式为 $f(x)=(13+10x+2x^2) \bmod 17$。分别取 $x=1,2,3,4,5$,试计算出对应的 5 个子密钥,并根据这 5 个子密钥的任意 3 个恢复出秘密 s。

5-14 设 $p=17$,$k=3$,$n=5$,令 $x_i=i(1\leqslant i\leqslant 5)$。假定 $f(1)=8$,$f(3)=10$,$f(5)=11$,试按 Shamir 门限方案重构秘密信息 s。

5-15 在 Asmuth-Bloom 门限方案中,设 $k=2$,$n=3$,$q=5$,$m_1=7$,$m_2=9$,$m_3=11$,3 个子密钥分别是 6、3、4,试恢复出秘密信息 s。

5-16 证明 5.8 节群密钥生成协议的共享密钥为 $K=\alpha^{r_0r_1+r_1r_2+\cdots+r_{t-2}r_{t-1}+r_{t-1}r_0} \bmod p$。为什么说当协议执行完后,群组内所有合法用户都可以计算出相同的群密钥,而群组外的其他人不能获取任何有用的信息?

5-17 5.8 节的群密钥分发方案的中间人攻击是如何实施的?请给出一种改进的办法或思路。

5-18 PKI(Public Key Infrastructure)应用实践:目的是加深对 CA 认证原理及其结构的理解,掌握在 Windows Server 环境下独立根 CA 的安装和使用,掌握证书服务的管理,掌握基于 Web 的 SSL 连接设置,加深对 SSL 的理解。应用实践完成以下内容:

(1) CA 的使用:客户端通过 Web 页面申请证书;服务器端颁发证书;客户端证书的下载与安装。

(2) 证书服务的管理:停止/启动证书服务;CA 备份/还原;证书废除;证书吊销列表的创建与查看。

(3) 基于 Web 的 SSL 连接设置:为 Web 服务器申请证书并安装;在 Web 服务器端配置 SSL 连接;客户端通过 SSL 与服务器端建立连接。