第 3 章　公钥密码学

一般理解，密码学就是保护信息传递的机密性，但机密性仅仅是当今密码学一个主题。对信息发送方与接收方的身份验证、对所发出/接收信息在事后的不可抵赖以及保障数据的完整性是现代密码学另外一些重要主题。公钥密码体制对这些问题都给出了出色的解答。

1976 年 Diffie 和 Hellman 发表了“密码学的新方向”一文，提出了公开密钥密码体制（简称公钥密码体制）的思想，奠定了公钥密码学的基础。公钥密码体制是现代密码学的最重要的发明和进展，开创了密码学的新时代。

传统的对称密码体制中，加密和解密使用相同的密钥，每对用户之间都需要共享一个密钥，而且需要保持该密钥的机密性。当通信的用户数目比较多的时候，密钥的产生、存储和分发是一个很大的问题。而公钥密码体制则将加密密钥、解密密钥甚至加密算法、解密算法分开，用户只需掌握解密密钥，而将加密密钥和加密函数公开。任何人都可以加密，但只有掌握解密密钥的用户才能解密。公钥密码体制从根本上改变了密钥分发的方式，给密钥管理带来了诸多便利。公钥密码体制不仅用于加解密，而且可以广泛用于消息鉴别、数字签名和身份认证等服务，是密码学中一个开创性的成就。

公钥密码体制的最大优点是适应网络的开放性要求，密钥管理相对于对称密码体制要简单地多。但是，公钥密码体制并不会取代对称密码体制，原因在于公钥密码体制算法相对复杂，加解密速度较慢。实际应用中，公钥密码和对称密码经常结合起来使用，加解密使用对称密码技术，而密钥管理使用公钥密码技术。

3.1　公钥密码体制原理

从密码学产生至 20 世纪 70 年代公钥密码产生之前，传统密码体制，包括古典密码和现代对称密码，都是基于替换和置换这些初等方法。公钥密码学与其前的密码学完全不同。首先，公钥算法建立在数学函数基础上，而不是基于替换和置换。其安全性基于数学上难解的问题，如大整数因子分解问题、有限域的离散对数问题、平方剩余问题、椭圆曲线的离散对数问题等。其次，与只使用一个密钥的传统密码技术不同，公钥密码学是非对称的，加/解密分别使用两个独立的密钥：加密密钥可对外界公开，称为公开密钥或公钥；解密密钥只有所有者知道，称为秘密密钥或私钥。公钥和私钥之间具有紧密联系，用公钥加密的信息只能用相应的私钥解密，反之亦然。要想由一个密钥推知另一个密钥，在计算上是不可能的。基于公钥密码体制，通信双方无需预先商定密钥就可以进行秘密通信，克服了对称密码体制中必须事先使用一个安全通道约定密钥的缺点。

3.1.1 公钥密码体制

公钥密码算法依赖于一个加密密钥和一个与之相关的不同的解密密钥，这些算法都具有下述重要特点：

- 加密/解密算法相同，但使用不同的密钥。
- 发送方拥有加密密钥或解密密钥，而接收方拥有另一个密钥。
- 根据密码算法和加密密钥以及若干密文，要恢复明文在计算上是不可行的。
- 根据密码算法和加密密钥，确定对应的解密密钥在计算上是不可行的。

公钥密码体制有 6 个组成部分，如图 3-1 所示。

(1) 明文。算法的输入。它们是可读信息或数据。

(2) 加密算法。加密算法对明文进行各种转换。

(3) 公钥和私钥。算法的输入。这对密钥中一个用于加密，一个用于解密。加密算法执行的变换依赖于公钥和私钥。

(4) 密文。算法的输出。它依赖于明文和密钥，对给定的消息，不同的密钥产生的密文不同。

(5) 解密算法。该算法接收密文和相应的密钥，并产生原始的明文。

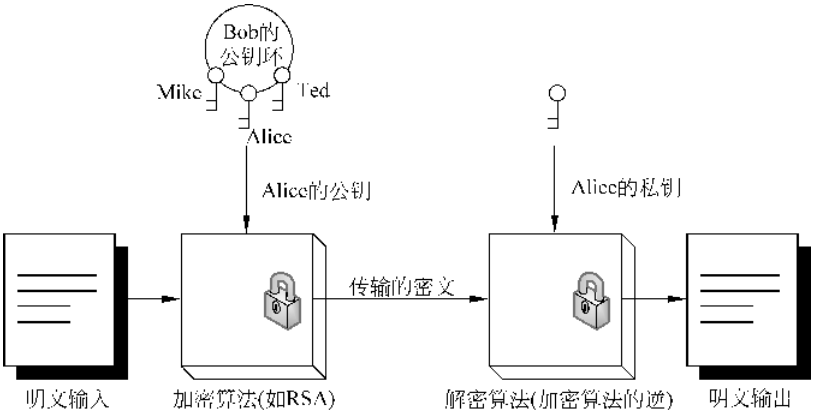

图 3-1 公钥密码体制

公钥密码体制的主要工作步骤包括：

① 每一用户产生一对密钥，分别用来加密和解密消息。

② 每一用户将其中一个密钥存于公开的寄存器或其他可访问的文件中，该密钥称为公钥。另一密钥是私有的。任一用户可以拥有若干其他用户的公钥。

③ 发送方用接收方的公钥对消息加密。

④ 接收方收到消息后，用其私钥对消息解密。由于只有接收方知道其自身的私钥，所以其他的接收者均不能解密出消息。

利用这种方法，通信各方均可访问公钥，而私钥是各通信方在本地产生的，所以不必进行分配。只要用户的私钥受到保护，保持秘密性，那么它的通信就是安全的。在任何时刻，系统可以改变其私钥，并公布相应的公钥以替代原来的公钥。

表 3-1 总结了对称密码和公钥密码的一些重要特征。

表 3-1　对称密码和公钥密码

密码类型	对称密码	公钥密码
一般要求	加密和解密使用相同的密钥	同一算法用于加密和解密，但加密和解密使用不同密钥
	收发双方必须共享密钥	发送方拥有加密或解密密钥，而接收方拥有另一密钥
安全性要求	密钥必须是保密的	两个密钥之一必须是保密的
	若没有其他信息，则解密消息是不可能或至少是不可行的	若没有其他信息，则解密消息是不可能或至少是不可行的
	知道算法和若干密文不足以确定密钥	知道算法和其中一个密钥以及若干密文不足以确定另一密钥

公钥密码的两种基本用途是用来进行加密和认证。不妨假设消息的发送方为 A，相应的密钥对为(PU_A，PR_A)，其中 PU_A 表示 A 的公钥，PR_A 表示 A 的私钥。同理，假设消息的接收方为 B，相应的密钥对为(PU_B，PR_B)，其中 PU_B 表示 B 的公钥，PR_B 表示 B 的私钥。现 A 欲将消息 X 发送给 B。A 从自己的公钥环中取出接收方 B 的公钥 PU_B，对作为输入的消息 X 和加密密钥 PU_B，A 生成密文 Y：

$$Y = E(PU_B, X)$$

B 收到加密消息后，用自己的私钥 PR_B 对密文进行解密，恢复明文 X：

$$X = D(PR_B, Y)$$

整个过程如图 3-2 所示。

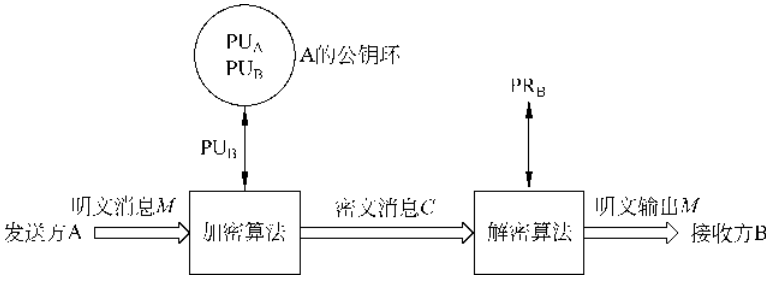

图 3-2　公钥密码用于保密

由于 A 是用 B 的公钥 PU_B 对消息进行加密，因此只有用 B 的私钥 PR_B 才能解密密文 Y，而 B 的私钥 PR_B 是由 B 秘密保存的。由于攻击者没有 B 的私钥 PR_B，因此攻击者仅根据密文 C 和 B 的公钥 PU_B 解密消息是不可能的。由此，就实现了保密性的功能。

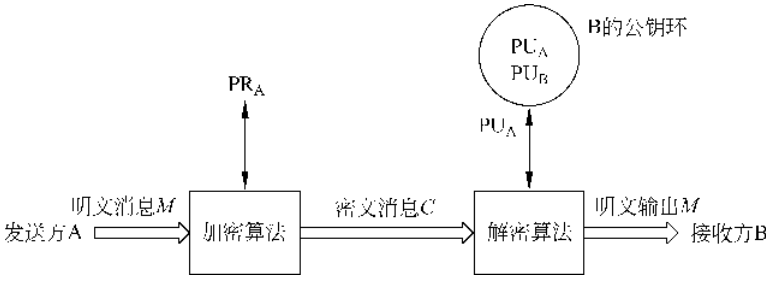

图 3-3　公钥密码用于认证

除了用于实现保密性之外，公钥密码还可以用来实现认证功能，实现过程如图 3-3 所示。在这种方法中，A 向 B 发送消息前，先用 A 的私钥 PR_A 对消息 X 加密：

$$Y = E(PR_A, X)$$

B 则用 A 的公钥 PU_A 对消息解密：

$$X = D(PU_A, Y)$$

由于只有发送方 A 拥有私钥 PR_A，因此只要接收方 B 能够正确解密密文 Y，就可以认为消息的确是由发送方 A 发出的。这样就实现了对发送方 A 的身份认证。

上述方法是对整条消息加密，尽管这种方法可以验证发送方和消息的有效性，但却需要大量的存储空间。在实际使用中，只对一个称为认证符的小数据块加密，它是该消息的函数，对该消息的任何修改必然会引起认证符的变化。

在图 3-3 所示认证过程中，由于攻击者也可以知道 A 的公钥，因此攻击者也可以解密密文消息 Y。也就是说，这里只能实现认证能力，而无法实现保密能力。如果要同时实现保密和认证功能，需要对消息进行两次加密，如图 3-4 所示。

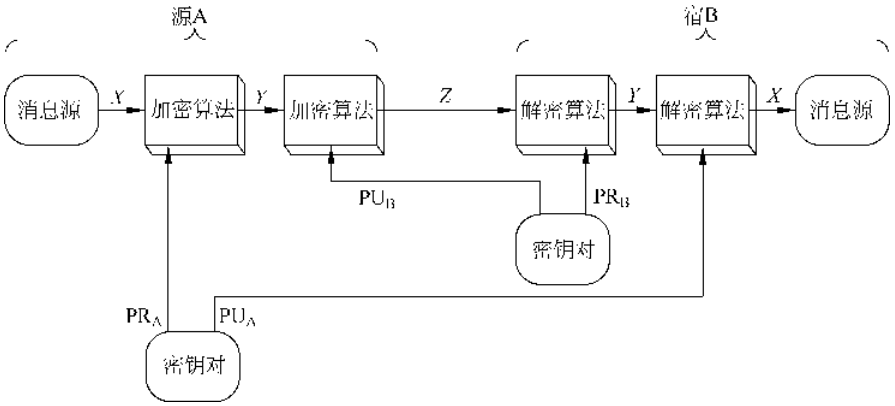

图 3-4 公钥密码用于保密和认证

在这种方法中，发送方首先用其私钥对消息加密，得到数字签名，然后再用接收方的公钥加密：

$$Z = E(PU_B, E(PR_A, X))$$

所得的密文只能被拥有相应私钥的接收方解密：

$$X = D(PU_A, D(PR_B, Z))$$

这种方式既可实现消息的保密性，又可以实现对发送方的身份认证。但这种方法的缺点是，在每次通信中要执行四次复杂的公钥算法。

3.1.2 对公钥密码的要求

Diffie 和 Hellman 给出了公钥密码体制应满足的 5 个基本条件：

(1) 产生一对密钥(公钥 PU，私钥 PR)在计算上是容易的。

(2) 已知接收方 B 的公钥 PU_B 和要加密的消息 M，消息发送方 A 产生相应的密文在计算上是容易的：

$$C = E(PU_B, M)$$

(3) 消息接收方 B 使用其私钥对接收的密文解密以恢复明文在计算上是容易的：

$$M = \mathrm{D}(\mathrm{PR_B}, C) = \mathrm{D}[\mathrm{PR_B}, \mathrm{E}(\mathrm{PU_B}, M)]$$

(4) 已知公钥 $\mathrm{PU_B}$ 时，攻击者要确定对应的私钥 $\mathrm{PR_B}$ 在计算上是不可行的。

(5) 已知公钥 $\mathrm{PU_B}$ 和密文 C，攻击者要恢复明文 M 在计算上是不可行的。

有研究者认为还可以增加下面一个附加条件。

加密和解密函数的顺序可以交换，即：

$$M = \mathrm{D}[\mathrm{PU_B}, \mathrm{E}(\mathrm{PR_B}, M)] = \mathrm{D}[\mathrm{PR_B}, \mathrm{E}(\mathrm{PU_B}, M)]$$

例如，著名的 RSA 密码就满足上述附加条件。但是，这一条件并不是必须的，不是所有的公钥密码应用都满足该条件。

在公钥密码学概念提出后的几十年中，只有两个满足这些条件的算法(RSA，椭圆曲线密码体制)为人们普遍接受，这一事实表明要满足上述条件是不容易的。这是因为，公钥密码体制是建立在数学中的单向陷门函数的基础之上的。

单向函数是满足下列性质的函数：每个函数值都存在唯一的逆；对定义域中的任意 x，计算函数值 $f(x)$是非常容易的；但对 f 的值域中的所有 y，计算 $f^{-1}(y)$在计算上是不可行的，即求逆是不可行的。

一个单向函数，如果给定某些辅助信息(称为陷门信息)，就易于求逆，则称这样的单向函数为一个陷门单向函数。即单向陷门函数是满足下列条件的一类可逆函数 f_k：

- 若 k 和 X 已知，则容易计算 $Y=f_k(X)$。
- 若 k 和 Y 已知，则容易计算 $X=f_k^{-1}(Y)$。
- 若 Y 已知但 k 未知，则计算出 $X=f_k^{-1}(Y)$是不可行的。

公钥密码体制就是基于这一原理，将辅助信息(陷门信息)作为私钥而设计的。这类密码的安全强度取决于它所依据的问题的计算复杂度。由此可见，寻找合适的单向陷门函数是公钥密码体制应用的关键。目前比较流行的公钥密码体制主要有两类：一类是基于大整数因子分解问题的，最典型的代表是 RSA；另一类是基于离散对数问题的，例如椭圆曲线公钥密码体制。

3.2　RSA 算法

MIT 的 Ron Rivest，Adi Shemir 和 Len Adleman 于 1978 在题为《获得数字签名和公开钥密码系统的方法》的论文中提出了基于数论的非对称密码体制，称为 RSA 密码体制。RSA 算法是最早提出的满足要求的公钥算法之一，也是被广泛接受且被实现的通用公钥加密方法。

RSA 是一种分组密码体制，其理论基础是数论中“大整数的素因子分解是困难问题”的结论，即求两个大素数的乘积在计算机上时容易实现的，但要将一个大整数分解成两个大素数之积则是困难的。RSA 公钥密码体制安全、易实现，是目前广泛应用的一种密码体制，既可以用于加密，又可以用于数字签名。

3.2.1　算法描述

RSA 明文和密文均是 0～n－1 之间的整数，通常 n 的大小为 1024 位二进制数，即 n 小

于 2^{1024}。

1. 密钥生成

首先必须生成一个公钥和对应的私钥。选择两个大素数 p 和 q(一般约为 256 比特),p 和 q 必须保密。计算这两个素数的乘积 $n=p\times q$,并根据欧拉函数计算小于 n 且与 n 互素的正整数的数目:

$$\phi(n)=(p-1)(q-1)$$

随机选择与 $\phi(n)$互素的数 e,则得到公钥$<e,n>$。计算 $e \bmod \phi(n)$的乘法逆 d,即 d 满足:

$$e\times d\equiv 1(\bmod \phi(n))$$

则得到了私钥$<d,n>$。

2. 加密运算

在 RSA 算法中,明文以分组为单位进行加密。将明文消息 M 按照 n 比特长度分组,依次对每个分组做一次加密,所有分组的密文构成的序列即是原始消息的密文 C。加密算法如下:

$$C=M^e \bmod n$$

其中收发双方均已知 n,发送方已知 e,只有接收方已知 d。

3. 解密运算

解密算法如下:

$$M=C^d \bmod n=(M^e)^d \bmod n=M^{ed} \bmod n$$

图 3-5 归纳总结了 RSA 算法。

密钥产生	
选择 p 和 q	p 和 q 都是素数,$p\neq q$
计算 $n=p\times q$	
计算 $\phi(n)=(p-1)(q-1)$	
选择整数 e	$\gcd(\phi(n),e)=1$; $1<e<\phi(n)$
计算 d	$d\equiv e^{-1}(\bmod \phi(n))$
公钥	PU=$\{e,n\}$
私钥	PR=$\{d,n\}$

加密	
明文:	$M<n$
密文:	$C=M^e \bmod n$

解密	
密文:	C
明文:	$M=C^d \bmod n$

图 3-5 RSA 算法

RSA 的缺点主要有以下两点:

(1) 产生密钥很麻烦,受到素数产生技术的限制,因而难以做到一次一密。

(2) 分组长度太大,为保证安全性,n 至少也要 600 位以上,使运算代价很高,尤其是速度较慢,较对称密码算法慢几个数量级;且随着大数分解技术的发展,这个长度还在增加,

不利于数据格式的标准化。因此，一般来说 RSA 只用于少量数据加密。

3.2.2　RSA 的安全性

1. 因子分解

RSA 算法的安全性是建立在“大整数因子分解困难”这一事实上。由算法过程可以看出，分解 n 与求 $\phi(n)$等价，若分解出 n 的因子，则 RSA 算法将变得不安全。因此分解 n 是最明显的攻击方法。

利用因子分解进行的攻击主要有如下几种具体作法：

(1) 分解 n 为两个素因子 $p\times q$。这样就可以计算出 $\phi(n)=(p-1)(q-1)$，从而可以计算出 $d\equiv e^{-1}(\bmod \phi(n))$。

(2) 直接确定 $\varphi(n)$而不先确定 p 和 q。这同样也可以确定 $d\equiv e^{-1}(\bmod \phi(n))$。

对 RSA 的密码分析的讨论大都集中于第一种攻击方法，即将 n 分解为两个素数因子从而计算出私钥。RSA 的安全性依赖于大数分解，但是否等同于大数分解一直未能得到理论上的证明，因为没有证明破解 RSA 就一定需要作大数分解。目前，RSA 的一些变种算法已被证明等价于大数分解。不管怎样，分解 n 是最明显的攻击方法，大量的数学高手也试图通过这个途径破解 RSA，但至今一无所获。因此，从经验上讲，RSA 是安全的。

但需要注意的是，尽管因子分解具有大素数因子的数 n 仍然是一个难题，但已不像以前那么困难。计算能力的不断增强和因子分解算法的不断改进，给大密钥的使用造成威胁。因此我们在选择 RSA 的密钥大小时必须选大一些，一般而言取在 1024～2048 位，具体大小视应用而定。

为了防止可以很容易地分解 n，RSA 算法的发明者建议 p 和 q 还应满足下列限制条件：

(1) p 和 q 的长度应仅相差几位。这样对 1024 位的密钥而言，p 和 q 都应约在 $10^{75}\sim 10^{100}$之间。

(2) $(p-1)$和$(q-1)$都应有一个大的素因子。

(3) $\gcd(p-1,q-1)$应该较小。

另外，已经证明，若 $e<n$ 且 $d<n^{1/4}$，则 d 很容易被确定。

2. 选择密文攻击

RSA 在选择密文攻击面前很脆弱。一般攻击者是将某一信息作一下伪装，让拥有私钥的实体签署。然后，经过计算就可得到它所想要的信息。

例如，Eve 在 Alice 的通信过程中进行窃听，获得了一个用她的公开密钥加密的密文 C，并试图恢复明文。从数学上讲，即计算 $m=C^d \bmod n$。为了恢复 m，Eve 首先选择一个随机数 $r(r<n)$，然后计算：

$$x = r^e \bmod n,\quad y = xC \bmod n$$

以及 $r \bmod n$ 的乘法逆 t，即 t 满足

$$t\times r = 1 \bmod n$$

现在 Eve 想方设法让 Alice 用她的私钥对 y 整体签名：

$$u = y^d \bmod n$$

因为 $r=x^d \bmod n$，所以 $r^{-1}x^d \bmod n=1$，通过计算

$$t \times u \bmod n = r^{-1} y^d \bmod n = r^{-1}\ x^d\ C^d \bmod n = C^d \bmod n = m$$

Eve 就轻松得获得 Alice 发的明文 m 了。

实际上，攻击利用的都是同一个弱点，即存在这样一个事实：乘幂保留了输入的乘法结构：

$$(X \times M)^d = X^d \times M^d \bmod n$$

这个固有的问题来自于公钥密码系统的最有用的特征：每个人都能使用公钥。从算法上无法解决这一问题，主要措施有两条：一条是采用好的公钥协议，保证工作过程中实体不对其他实体任意产生的信息解密，不对自己一无所知的信息签名；另一条是决不对陌生人送来的随机文档签名，签名时首先对文档作 Hash 处理，或同时使用不同的签名算法。

3.3 ElGamal 公钥密码体制

ElGamal 公钥密码体制是由 ElGamal 于 1985 年提出来的，是一种基于离散对数问题的密码体制。ElGamal 既可以用于加密，又可以用于签名，是 RSA 之外最有代表性的公钥密码体制之一，并得到了广泛的应用。数字签名标准 DSS 就是采用了 ElGamal 签名方案的一种变形。

1. 密钥生成

首先选择一个大素数 p，并要求 p 有大素数因子。Z_p 是一个有 p 个元素的有限域，Z_p^* 是 Z_p 中非零元构成的乘法群，$g \in Z_p^*$ 是一个本源元。然后选择随机数 k，满足 $1 \leqslant k \leqslant p-1$。计算 $y = g^k \bmod p$，则公钥为 (y, g, k)，私钥为 k。

2. 加密算法

待加密的消息为 $M \in Z_p$。选择随机数 $r \in Z_{p-1}^*$，然后计算：

$$C_1 = g^r \bmod p$$

$$C_2 = My^r \bmod p$$

则密文 $C = (C_1, C_2)$。

3. 解密算法

收到密文 $C = (C_1, C_2)$ 后，执行以下计算：

$$M = C_2 / C_1^k \bmod p$$

则消息 M 被恢复。

4. ElGamal 安全性

ElGamal 密码体制的安全性基于有限域 Z_p 上的离散对数问题的困难性。目前，尚没有求解有限域 Z_p 上的离散对数问题的有效算法。所以当 p 足够大时（一般是 160 位以上的十进制数），ElGamal 密码体制是安全的。

此外，加密中使用了随机数 r。r 必须是一次性的，否则攻击者获得 r 就可以在不知道私钥的情况下加密新的密文。

3.4　密 钥 管 理

随着计算机网络的发展，人们对网络上传递敏感信息的安全性要求也越来越高，密码技术得到了广泛应用。随之而来的，如何生成、分发、管理密钥也成为一个重要的问题。密钥管理的核心问题是：确保使用中的密钥能安全可靠。

根据应用场合的不同，密钥可以分成以下几类。

工作密钥，也叫基本密钥或初始密钥。由用户选定或系统分配，使用期限一般较长，如数月甚至一年等。

会话密钥，即通信双方交换数据时使用的密钥。会话密钥一般由通信双方协商决定，也可由密钥分配中心分配。会话密钥大多是临时的、动态的，可以降低密钥的分配和存储的数目。

密钥加密密钥，主要用于对要传送的会话密钥进行加密，也叫做二级密钥。

主机主密钥，对应于层次化密钥管理结构中的最顶层，主要用于对密钥加密密钥进行加密保护，一般保存于主结点，受到严格保护。

公钥密码体制的主要作用之一就是解决密钥分配问题。公钥密码可用于下列两个不同的方面：

(1) 公钥密码体制中的公钥分配。

(2) 对称密码体制中的密钥分配。

3.4.1　公钥分配

人们已经提出了几种公钥分配方法，所有这些方法本质上可归结为下列几种方法：

- 广播式公钥分发；
- 目录式公钥分发；
- 公钥授权；
- 公钥证书。

1. 广播式的公钥发布

公钥密码算法的特点就是公钥可以公开，因此如果有像 RSA 这样为人们广泛接受的公钥算法，那么任一通信方可以将他的公钥发送给另一通信方或广播给通信各方。例如，用于邮件安全的 PGP 就是在消息后面附上公钥，并将其发送到网络上。虽然这种方法比较简便，但它有一个较大的缺点，即任何人都可以伪造这种公钥的公开发布。也就是说，某个用户可以假冒是用户 A 并将一个公钥发送给通信的另一方或广播该公钥，在用户 A 发现这种假冒并通知其他各方之前，该假冒者可以读取所有本应发送给 A 的加密后的消息，并且可以用伪造的密钥进行认证。因此，需要对收到的公钥进行鉴别。

2. 公开可访问的目录

由可信机构负责维护一个动态可访问的公钥的公开目录，这种方式可以获得更大程度的安全性，参见图 3-6。这种方法包含下面几方面的内容。

(1) 可信机构通过对每一通信方建立一个目录项<用户名,公钥>来建立、维护该公钥目录。

(2) 各通信方通过访问该目录来注册一个公钥。注册必须亲自或通过安全的认证通信来进行。

(3) 通信方可以随时访问该公钥目录,以及申请删除、修改、更新当前的公钥。这可能是因为公钥已用于大量的数据,因而用户希望更换公钥,也可能是因为相应的私钥已经泄密。

(4) 为安全起见,通信方和可信机构之间的通信受鉴别保护。

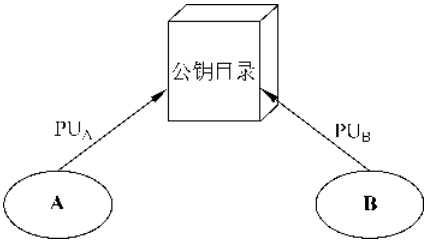

图 3-6 公开的公钥发布

这种方法显然比由个人公开发布公钥要安全,但是它也存在缺点。如果攻击者获得或计算出目录管理员的私钥,则他可以发布伪造的公钥,假冒任何通信方,以窃取发送给该通信方的消息。另外,攻击者也可以通过修改目录管理员保存的记录来达到这一目的。

3. 公钥授权

通过更加严格地控制目录中的公钥分配,可使公钥分配更加安全。图 3-7 举例说明了一个典型的公钥分配方案。像公钥授权方案一样,该方案假定由一个专门的权威机构负责维护一个包含所有通信方公钥的动态目录,除此之外,每一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相应的私钥。这种方案主要用于通信方 A 要与 B 通信时,向权威机构请求 B 的公钥,主要包含以下步骤(与图 3-7 中序号对应):

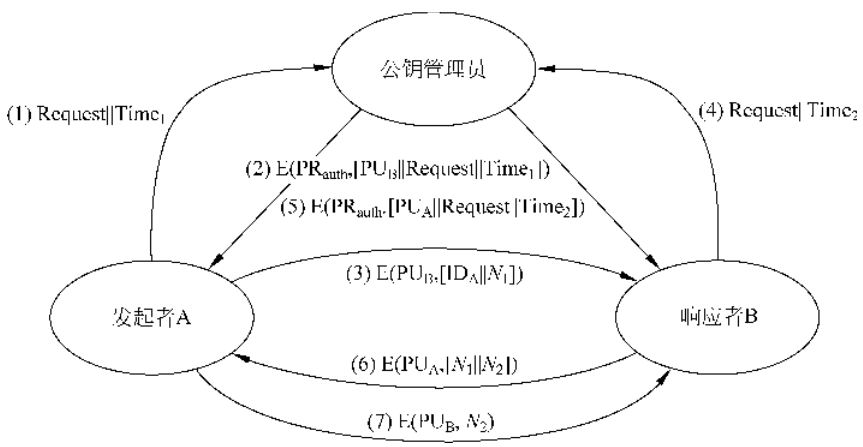

图 3-7 公钥授权

(1) A 发送一条带有时间戳的消息给目录管理员,以请求 B 的当前公钥。

(2) 管理员给 A 发送一条用其私钥 PR_{auth} 加密的消息,这样 A 就可用管理员的公钥对接收到的消息解密,因此 A 可以确信该消息来自管理员。这条消息包括下列内容:

- B 的公钥 PU_B。A 可用它对要发送给 B 的消息加密。
- 原始请求。这样 A 可以将该请求与其最初发出的请求进行比较,以验证在管理员收到请求之前,其原始请求未被修改。
- 原始时间戳。这样 A 可以确定它收到的不是来自管理员的旧消息,该旧消息中包含

的不是 B 的当前公钥。

(3) A 保存 B 的公钥,并用它对包含 A 的标识(ID_A)和临时交互号(N_1)的消息加密,然后发送给 B。这里,临时交互号是用来唯一标识本次交易的。

(4) 与 A 检索 B 的公钥一样,B 以同样的方法从管理员处检索出 A 的公钥。

至此 A 和 B 已安全地获得了彼此的公钥,双方的信息交换将受到保护。尽管如此,但是最好还包含下面两步:

(5) B 用 PU_B 对 A 的临时交互号(N_1)和 B 所产生的新临时交互号(N_2)加密,并发送给 A。因为只有 B 可以解密消息(3),所以消息(6)中的 N_1。可以使 A 确信其通信伙伴就是 B。

(6) A 用 B 的公钥对加 N_2 加密并发送给 B,以使 B 相信其通信伙伴是 A。

这样,总共需要发送 7 条消息。但是由于 A 和 B 可保存另一方的公钥以备将来使用(这种方法称为暂存),所以并不会频繁地发送前面 4 条消息。不过为了保证通信中使用的是当前公钥,用户应定期地申请对方的当前公钥。

4. 公钥证书

在公钥授权方案中,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,因此公钥管理员就会成为系统的瓶颈。像前面一样,目录管理员所维护的含有用户名和公钥的目录也容易被篡改。

公钥证书方法最早是由 Kohnfelder 提出的,目的是使得通信各方使用证书来交换公钥,而无需一个权威机构的在线服务。在某种意义上,这种方案与直接从权威机构处获得公钥的可靠性相同。公钥证书包含公钥和公钥拥有者的标识,并由可信的第三方进行签名。通常,第三方是一个权威机构,如政府机构,或者金融机构,为整个用户群所信任。一个用户以一种安全的方式将他的公钥交给权威机构的公钥管理员,从而获得一个证书,并公开自己的公钥证书。任何需要该用户公钥的人都可以获得这个证书,并通过查看附带的权威机构的签名来验证证书的有效性。通信一方也可以通过传递证书的方式将他的密钥信息传达给另一方。这种方法应满足下列要求:

(1) 任何通信方可以读取证书并确定证书拥有者的身份和公钥。

(2) 任何通信方可以验证该证书是否由权威机构签发,以及是否有效。

(3) 只有权威机构才可以签发并更新证书。

图 3-8 举例说明了证书方法。每一通信方向权威机构的证书管理员提供一个公钥,并申请一个公钥证书。申请必须由当事人亲自或通过某种安全的认证通信提出。对于申请者 A,管理员提供如下形式的证书:

$$C_A = E(PR_{auth}, [T \| ID_A \| PU_A])$$

其中 PR_{auth} 是证书管理员的私钥,T 是时间戳。A 将该证书发送给其他通信各方,他们以如下方式来验证证书:

$$D(PR_{auth}, C_A) = D(PR_{auth}, E(PR_{auth}, [T \| ID_A \| PU_A])) = (T \| ID_A \| PU_A)$$

接收方用管理员的公钥 PU_{auth} 对证书解密。因为只用管理员的公钥才可读取证书,因此接收方可验证证书确实是出自证书管理员;ID_A 和 PU_A 向接收方提供证书拥有者的身份标识和公钥;时间戳 T 用来验证证书的当前性,抵抗攻击者的重放攻击。假设 A 的私钥泄露,产生新的公/私钥对并向证书管理员申请新的证书;而此时,攻击者重放 A 的旧证书给 B。若 B 用 A 的旧公钥加密消息,则攻击者可读取消息。