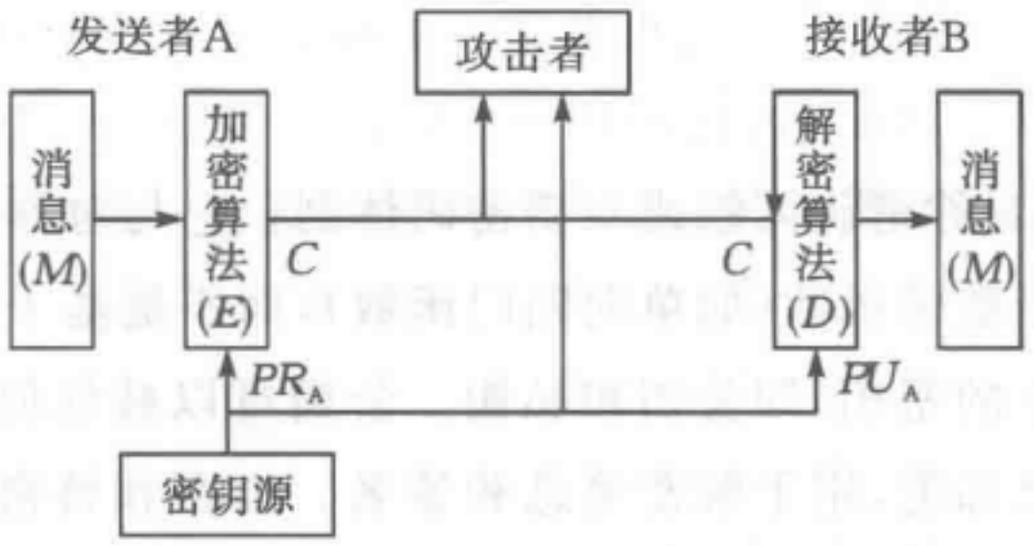

图 2-22 公钥认证模型

在上面的认证模型中，认证是对发送方的整个消息进行加密，这种方法可以验证发送者和消息的有效性，但却需要大量的储存空间。实际的做法是先对消息进行一个函数变换，将消息变换成一个小数据，然后再对小数据进行签名。

在认证模型中，消息没有保密，任何人都可以用发送者的公钥解密消息。如果综合加密模型和认证模型，则将同时具有保密和认证功能，如图 2-23 所示。

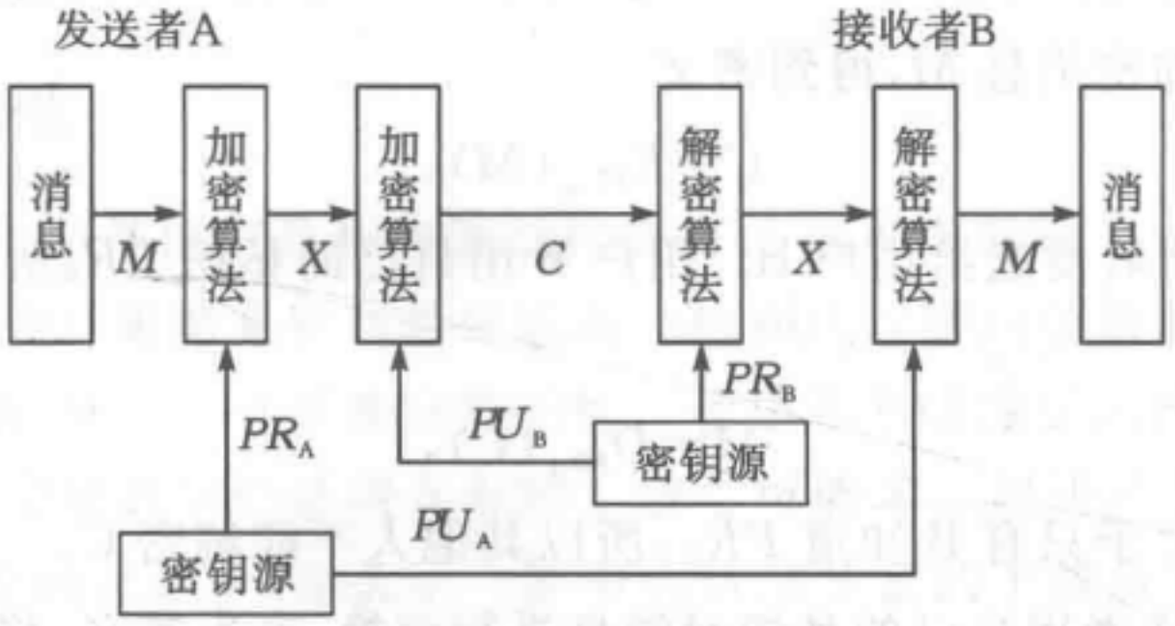

图 2- 23 公钥密码体制的保密和认证

发送者 A 先用自己的私钥 PR_A 加密消息 M，用于提供数字签名。再用接收者 B 的公钥 PU_B 加密，表示为：

$$C=E_{PU_B}(E_{PR_A}(M))。$$

接收者 B 在解密时，先用自己的私钥解密，然后再用发送者 A 的公钥解密，表示为：

$$M=D_{PU_A}(D_{PR_B}(C))。$$

从上面的加密和认证模型中，我们可以发现一个公钥密码系统应该满足下面几个要求。

(1)同一算法用于加密和解密，但加密和解密使用不同的密钥。

(2)两个密钥中的任何一个都可用来加密，另一个用来解密，加密和解密次序可以交换。

(3)产生一对密钥(公钥和私钥)在计算上是可行的。

(4)已知公钥和明文，产生密文在计算上是容易的。

(5)接收者利用私钥来解密密文在计算上是可行的。

(6)仅根据密码算法和公钥来确定私钥在计算上是不可行的。

(7)已知公钥和密文，在不知道私钥的情况下，恢复明文在计算上是不可行的。

上面几个要求的实质是要找一个单向陷门函数。单向陷门函数是指计算函数值是容易的，而计算函数的逆是不可行的。陷门单向函数则存在一个附加信息，当不知道该附加信息时，求函数逆是困难的，但当知道该附加信息时，求函数逆就变得容易了。陷门单向函数在附

加信息未知时是单向函数，而当附加信息已知时，就不再是单向函数了。通常把附加信息称为陷门信息，将陷门信息作为私钥，公钥密码体制就是基于这一原理而设计的，其安全强度取决于它所依据的问题的计算复杂度。

2.5.2　公钥密码分析

和对称密码体制一样，如果密钥太短，公钥密码体制也易受到穷举攻击，因此密钥必须足够长。然而又由于公钥密码体制所使用的可逆函数的计算复杂性与密钥长度常常不是线性关系，而是比线性函数增大更快的函数，所以密钥长度太大又会使得加密和解密运算太慢而不实用。目前提出的公钥密码体制的密钥长度已经足够抵抗穷举攻击，但也使它的加密和解密速度变慢，因此公钥密码体制一般用于加密小数据，如会话密钥，目前主要用于密钥管理和数字签名。

对公钥密码算法的第二种攻击就是从公钥计算出私钥。到目前为止，还没有在数学上证明该方法不可行。

还有一种仅适用于对公钥密码算法的攻击法，称为穷举消息攻击。由于公钥密码算法常常用于加密短消息，只要穷举这些短消息，就可以解密消息。例如，假设用公钥算法加密 DES 的 56 位密钥，攻击者可以用算法的公钥对所有可能的 56 位密钥加密，再与截获的密文相比较，如果一样，则相应的明文即 DES 的密钥。因此不管公钥算法的密钥多长，这种攻击的本质是对 56 位 DES 密钥的穷举攻击，抵抗这种攻击的方法是在要发送的消息后面加一些随机位。

2.6　RSA 算法

RSA 算法是 1977 年由 Rivest、Shamir 和 Adleman 提出的非常著名的公钥密码算法，它基于大合数的质因子分解问题的困难性。RSA 算法是一种分组密码，明文和密文是 $0 \sim n-1$ 之间的整数，通常 n 的大小为 1024 位二进制数或 309 位十进制数。

2.6.1　RAS 算法描述

1. 密钥的产生

(1)随机选择两个大素数 p 和 q。

(2)计算 $n=p\times q$。

(3)计算秘密的欧拉函数 $\varphi(n)=(p-1)(q-1)$。

(4)选择 e 使得 $1<e<\varphi(\mathrm{n})$，且 $\gcd(e,\varphi(\mathrm{n}))=1$。

(5)解方程求出 d：

$$ed\equiv 1 \bmod \varphi(n)，且\ 0\leqslant d\leqslant n。$$

(6)公开公钥：$PU=\{e,n\}$。

(7)保存私钥：$PR=\{d,p,q\}$。

2. 加密过程

加密时明文以分组为单位进行加密，每个分组 m 的二进制值均小于 n，对明文分组 m 做加密运算：$C=m^e \bmod n$，且 $0 \leqslant m < n$。

3. 解密过程

密文解密 $M=c^d \bmod n$

假设选择素数：$p=47$ 和 $q=71$。

计算 $n=p*q=47\times71=3337$，$\varphi(n)=(p-1)(q-1)=46\times70=3220$。

选择 e：使 $\gcd(e,3220)=1$，选取 $e=79$。

决定 d：$ed \equiv 1 \bmod 3220$，得 $d=1019$。

公开公钥{79,3337}，保存私钥(1019,47,71}。

现假设消息为 $M=6882326879663$，进行分组，分组的位数比 n 要小，我们选取 $M_1=688$，$M_2=232$，$M_3=687$，$M_4=966$，$M_5=003$。

M_1 的密文为 $C_1=688^{79} \bmod 3337=1570$，继续进行类似计算，可得到最终密文为：

$C=1570275620912276158$

如果解密，计算 $M_1=1570^{1019} \bmod 3337=688$，类似可以求出其他明文。

2.6.2 RSA 算法的安全性

RSA 算法的安全性基于分解大整数的困难性假设。RSA 算法的加密函数 $c=m^e \bmod n$ 是一个单向函数，所以对于攻击者来说，试图解密密文在计算上是不可行的。对于接收方解密密文的陷门是分解 $n=p\times q$，由于接收者知道这个分解，它可以计算 $\varphi(n)=(p-1)(q-1)$，然后用扩展欧几里得算法来计算解密私钥 d，因此对 RSA 算法的攻击有下面几个方法。

1. 穷举攻击

最基本的攻击是穷举攻击，也就是尝试所有可能的私钥。抵抗穷举攻击的方法是使用大的密钥空间，所以位数越多越安全，但也增加了加密和解密的复杂性，因此密钥越大，系统运行速度也越慢。

2. 数学攻击

另一种攻击方式是数学攻击，它的实质是试图对两个素数乘积的分解，数学攻击主要采用下面的几种形式。

(1)直接将 n 分解为两个素数因子，这样就可以计算 $\varphi(n)=(p-1)(q-1)$，然后可以确定私钥 $d \equiv e^{-1} \bmod \varphi(n)$。

(2)在不事先确定 p 和 q 的情况下直接确定 $\varphi(n)$，同样可以确定 $d \equiv e^{-1} \bmod \varphi(n)$。

(3)不先确定 $\varphi(n)$ 而直接确定 d。

目前大部分关于 RSA 密码分析的讨论都集中在进行素因子分解上，给定 n 确定 $\varphi(n)$ 就等价于对 n 进行因子分解，给定 e 和 n 时使用目前已知算法求出 d，在时间开销上至少和因子分解问题一样大，因此可以把因子分解的性能作为一个评价 RAS 安全性的基准。

对大整数分解的威胁除了人类的计算能力外，还会来自分解算法的进一步改进。一直以来因子分解攻击都采用所谓二次筛的方式，最新的攻击算法是广义素数筛(GNFS)。该算法分解大数的性能被大大提高。由于大数分解近年来取得很大进展，因此，就目前来说，RSA 的