第八章 数字签名

8.1 数字签名的基本原理

8.1.1 数字签名的概念与特点

- 数据摘要。散列函数对消息处理产生的散列值，也称其为消息的散列值，摘要信息在数字签名中应用过程可以概述为：首先使用某种散列算法，对要发送的数据进行处理，生成数据摘要信息；然后采用公钥密码算法，用私钥加密数据摘要信息。
- 一个签名体制一般包括两个部分。一是发送方的签名部分，对消息M签名，可以记作S=Sig(K,M)，签名算法使用的密钥是秘密的，即签字者的私钥；二是接收方的认证部分，对签名S的验证可以记作Ver(M,S,K)—{真，假}，认证算法使用的密钥是发送方（即签名者）的公钥。

8.1.1 数字签名的概念与特点

- 数字签名的特点

（1）信息是由签名者发送的；

（2）信息自签发后到收到为止未曾做过任何修改；

（3）如果A否认对信息的签名，可以通过仲裁解决A和B之间的争议；

（4）数字签名不同于手写签名：数字签名随文本的变化而变化，手写签字反应某个人个性特征，是不变的；数字签名与文本信息是不可分割的，而用手写签字是附加在文本之后的，与文本信息是分离的。

8.1.1 数字签名的概念与特点

- 数字签名的形式化定义

“数字签名”系指在数据电文中，以电子形式所含、所附或在逻辑上与数据电文有联系的数据，和与数据电文有关的任何方法，它可用于数据电文有关的签字持有人和表明此人认可数据电文所含信息。

一个签名方案由签署算法与验证算法两部分构成，可用五元关系组(P,A,K,S,y)表示，其中，P是由一切可能消息(messages)所构成的有限集合；A是一切可能的签名的有限集合；K为有限密钥空间，是一些可能密钥的有限集合；任意k∈K，有签署算法Sigk∈S，Sigk：P→ A,对任意x∈P，有s=Sigk(x)，那么s∈S为消息x的签名，将(x,s)发送到签名验证者。对于密钥集合K，有对应的验证算法Verk∈y,满足：$Verk : P \times A \rightarrow \{真，假\}$

8.1.1 数字签名的概念与特点

- 数字签名的形式化定义

签名者收到(x,s)后，计算Verk(x,y)，若y=Sigk(x)，则Verk(x,y)为真；若y≠Sigk(x)，则Verk(x,y)为假。其中：①任意k∈K，函数Sigk和Verk都为多项式时间函数。②Verk为公开的函数，而Sigk为秘密函数。③如果坏人要伪造B对x的签名，再计算上不可能的。也即，给定x，仅有B能计算出签名y，使得Verk(x,y)=真。④一个签名方案不能是无条件安全的，有足够的时间，第三方总能伪造B的签名。

8.1.1 数字签名的概念与特点

- 数字签名的功能

（1）身份认证。收方通过发方的电子签名能够确认发方的确切身份，但无法伪造。

（2）保密。双方的通信内容高度保密，第三方无从知晓。

（3）完整性。通信的内容无法被篡改。

（4）不可抵赖。发方一旦将电子签字的信息发出，就不能再否认。

数字签名与数据加密完全独立。数据可以只签名或只加密，也可既签名又加密，当然，也可以既不签名也不加密。

8.1.2 数字签名方案的分类

1. 基于数学难题的分类
   （1）基于离散对数问题的签名方案
   （2）基于素因子分解问题的签名方案
   （3）上述两种的结合签名方案
2. 基于签名用户的分类
   （1）单个用户签名的数字签名方案
   （2）多个用户的数字签名方案
3. 基于数字签名所具有特性的分类
   （1）不具有自动恢复特性的数字签名方案
   （2）具有消息自动恢复特性的数字签名方案
4. 基于数字签名所涉及的通信角色分类
   （1）直接数字签名（仅涉及通信的源和目的两方）
   （2）需仲裁的数字签名（除通信双方外，还有仲裁方）

8.1.3 数字签名使用模式与使用原理

- 数字签名使用模式

（1）智慧卡式（2）密码式（3）生物测定式

- 数字签名使用原理

数字签名使用的是发送方的密钥对，发送方用自己的私有密钥进行加密，接收方用发送方的公开密钥进行解密。

这是一个一对多的关系：任何拥有发送方公开密钥的人都可以验证数字签名的正确性。而私有密钥的加密解密则使用的是接收方的密钥对，这是多对一的关系：任何知道接收方公开密钥的人都可以向接收方发送加密消息，只有唯一拥有接受方私有密钥的人才可以对信息解密。

通常一个用户拥有两个密钥对，一个密钥对用来对数字签名进行加密解密，另一个密钥对对私有密钥进行加密解密。这种方式提供了更高的安全性。

8.1.3 数字签名使用模式与使用原理

- 利用散列函数进行数字签名和验证的文件传输过程：

（1）发送方首先用哈希函数从原文得到数字摘要，然后采用公开密钥体系用发送方的私有密钥对数字摘要进行签名，并把签名后的数字摘要附在要发送的原文后面。

（2）发送方选择一个私密密钥对文件进行加密，并把加密后的文件通过网络传输到接收方。

（3）发送方用接收方的公开密钥对秘密密钥进行加密，并通过网络把加密后的秘密密钥传输到接收方。

（4）接收方使用自己的私有密钥对密钥信息进行解密，得到秘密密钥的明文。

（5）接收方用秘密密钥对文件进行解密，得到经过加密的数字摘要。

8.1.3 数字签名使用模式与使用原理

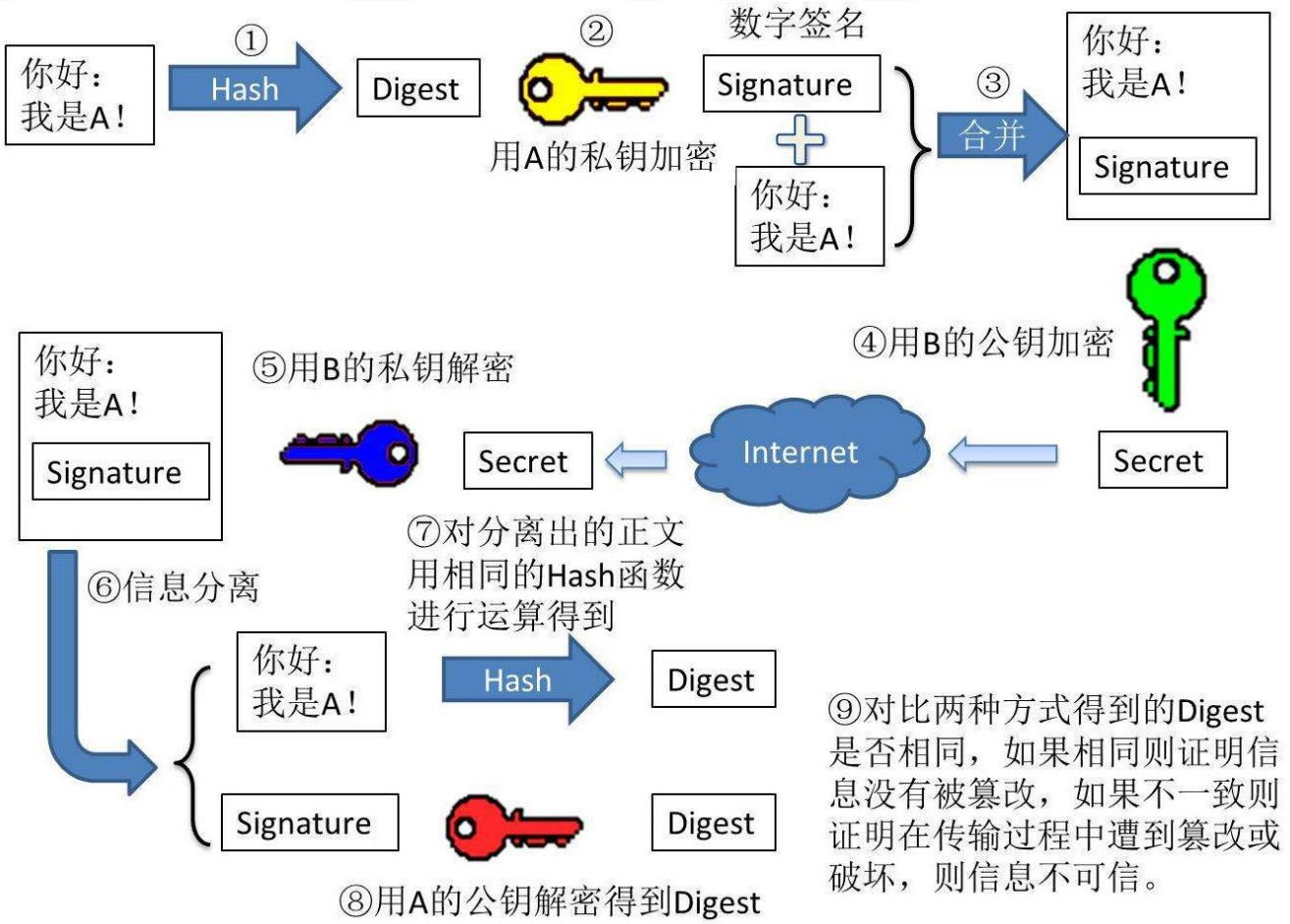
- 利用散列函数进行数字签名和验证的文件传输过程：

（6）接收方用发送方的公开密钥对数字签名进行解密，得到数字摘要的明文。

（7）接收方用得到的明文和哈希函数重新计算数字摘要，并与解密后的数字摘要进行对比。如果两个数字签名是相同的，说明文件在传输过程中没有被破坏。

8.1.3 数字签名使用模式与使用原理

- 利用散列函数进行数字签名和验证的文件传输过程：



8.2 RSA数字签名体制

8.2 RSA数字签名体制

- 数字签名过程：

（1）计算消息的散列值H(M)；

（2）用私钥(d,n)加密散列值：$s=(H(M))^d \bmod n$,签名结果就是s；

（3）发送消息和签名(M,s).当然，消息M很短的时候，可以直接对M用私钥加密，可表达为：$s=Sig(M)=M^d \bmod n$，签名时使用私钥(d,n)。

8.2 RSA数字签名体制

- 认证过程：

接收方收到(M,s)之后：

（1）取得发送方的公钥(e,n)；

（2）解密签名s：$h=s^e \bmod n$；

（3）计算消息的散列值H(M)；

（4）比较，如果h=H(M)，表示签名有效；否则，签名无效。

如果消息M很短的时候，可以直接对M用公钥解密以验证签名的有效性，可以表达为Ver(M,s)=真<=>$M=s^e \bmod n$

8.3 DSS数字签名体制

8.3 DSS数字签名体制

- 数字签名算法（DSA）：

(1)DSA算法参数说明：

DSA算法中应用了下述参数：

P：L bit长的素数。L是64的倍数，范围是512-1024；

Q：能被P-1整除的160bit的素数

G：$g = h^{(p-1)/q} \bmod p$，1< h < p - 1，g > 1；

X：x < q，x为私钥 ；

Y：y = g^x mod p ，(p，q，g，y)为公钥；

以上参数中p，q，g以及y为公匙，x为私匙必须保密!任何第三方用户想要从Y解密成X 都必须解决整数有限域离散对数难题。

8.3 DSS数字签名体制

- 数字签名算法（DSA）：

(2)签名：

P产生随机数k，$k < q$;

P计算 $r = (g^k \bmod p) \bmod q$

$s = (k^{-1} (H(m) + xr)) \bmod q$

签名结果是(m, r, s)。H(m)单向Hash函数。

(3)验证：

$w = s^{-1} \bmod q$

$u_1 = (H(m) * w) \bmod q$

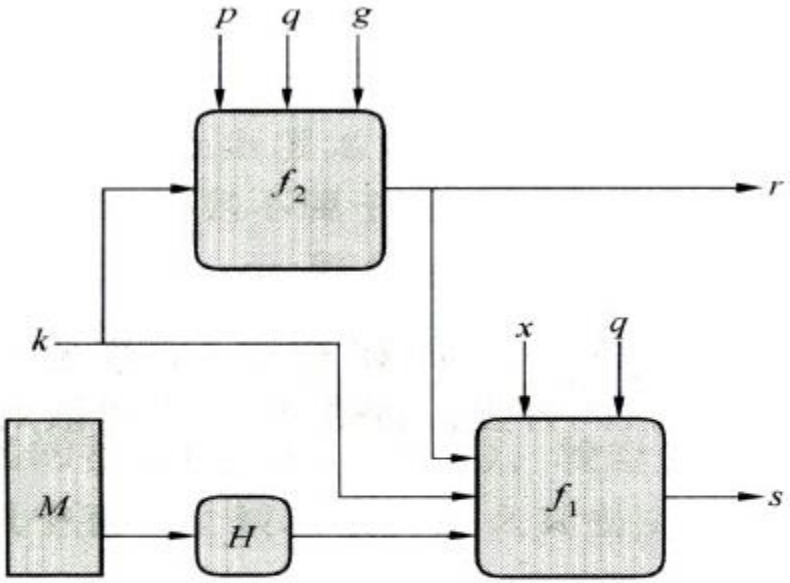
$u_2 = (r * w) \bmod q$

$v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$

若v = r，则认为签名有效。

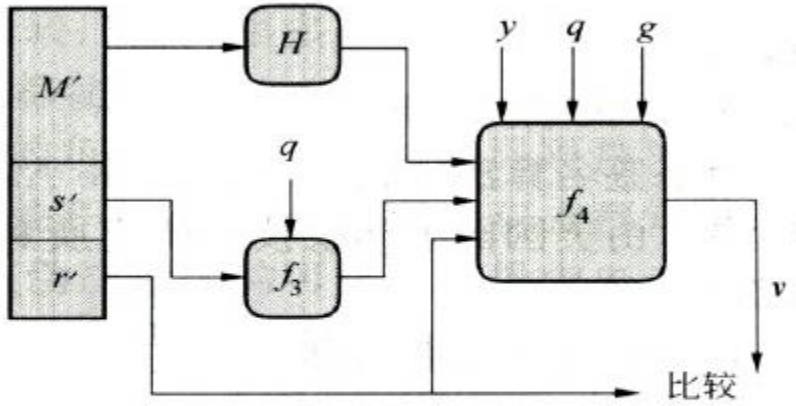
8.3 DSS数字签名体制

- DSS签名和验证函数：



$$s=f_1(H(M),k,x,r,q)=(k^{-1}(H(M)+xr)) \bmod q$$
$$r=f_2(k,p,q,g)=(g^k \bmod p) \bmod q$$

(a) 签名



$$w=f_3(s',q)=(s')^{-1} \bmod q$$
$$v=f_4(y,q,g,H(M')\ w,r')$$
$$=((g(H(M')w) \bmod q\ yr'w \bmod q) \bmod p) \bmod q$$

(b) 验证

8.3 DSS数字签名体制

- DSA的一个重要特点是两个素数公开，这样，当使用别人的p和q时，即使不知道私钥，也能确认它们是否是随机产生的。而RSA签名算法做不到。

问题：实现RSA数字签名和DSS数字签名算法。