计算机信息安全

– 数字签名技术

唐飞龙

Email：tang-fl@cs.sjtu.edu.cn

数字签名

- 数字签名概述
- 基于公钥密码体制的数字签名方案
- 特殊数字签名方案

数字签名概述

- 数字签名(协议)能够为消息发送方提供证明的一个过程 发送方提供证明的一个过程
- 数字签名实际上是一个把数字形式的消息和某个源发实体相联系的数据串: 把签名附加在一个消息或完全加密的消息上，以便于：
  - 消息的接收方能够鉴别消息的内容
  - 证明消息只能源发于所声称的发送方
- 数字签名的特点
  - A可通过签名向B证明自己就是A, 但B不能向任何人(甚至是自己)证明自己是A
- 数字签名协议由3个算法组成
  - 密钥生成算法：
    - 随机选择私钥，输出公钥、私钥对
  - 带有陷门的数字 签名算法（ Signature Algorithm ） ：
    - 给定消息和私钥，产生相应的签名
  - 验证算法（ Verification Algorithm ）：
    - 给定消息 公钥以 签名 输出接受/拒绝

◆其中，签名算法或签名密钥是秘密的，只有签名人知道，而验证算法是公开的

数字签名概述

- 数字签名的目的和要求

  数字签名的目的是保证信息的完整性和真实性，即消息内容没有被篡改，而且签名也没有被篡改，消息只能始发于所声称的发方。

- 数字签名任务：
  - 确认数据或程序的完整性： 验证消息在传输过程中有没有被篡改
  - 确认消息发送者的身份： 可以验证签名人,日期和时间
  - 出现纠纷可以由第三方仲裁
- 一个完善的签名方案应满足以下条件：
  - (1) 签名是可以被确认的，即收方可以确认或证实签名确实是由发方签名的；
  - (2) 签名是不可伪造的，即收方和第三方都不能伪造签名；
  - (3) 签名不可重用，即签名是消息(文件)的一部分，不能把签名移到其它消息(文件)上；
  - (4) 签名是不可抵赖的，即发方不能否认他所签发的消息；
  - (5) 第三方可以确认收发双方之间的消息传送但不能篡改消息。若当事双方对签名真伪发生争执时，能够在公正的仲裁者面前通过验证签名来确定其真伪

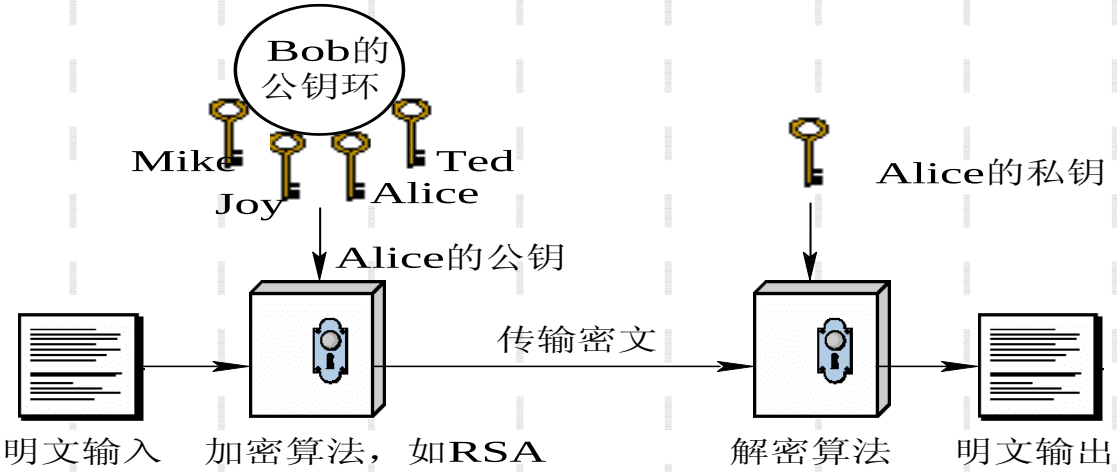
数字签名方案描述

- 数字签名方案的定义是：
  - 设M是消息的有限集合，S是签名的有限集合，K是密钥的有限集合，则数字签名算法sig是一个映射：
    sig：M×K→S，s = sig k (m)
  - 验证算法ver也是一个映射：

$$ver: M\times S\rightarrow\{True, False\},\ ver(m,s)=\begin{cases} True, & s=sig_k(m) \\ False, & s\neq sig_k(m) \end{cases}$$

  - 五元组{M，S，K，sig，ver}就称为一个签名方案。
- 归纳起来，一个数字签名方案的应用一般包括三个过程
  - （1）系统初始化过程：产生数字签名方案中用到的所有系统和用户参数，有公开的，也有秘密的
  - （2）签名产生过程：在此过程用户利用给定的签名算法和参数对消息产生签名，这种签名过程可以公开也可以不公开，但一定包含仅签名者才拥有的秘密信息（签名密钥）-只有签名者可以实施
  - （3）签名验证过程：验证者利用公开的验证方法和参数对给定消息的签名进行验证，得出签名的有效性-任何人都可以实施

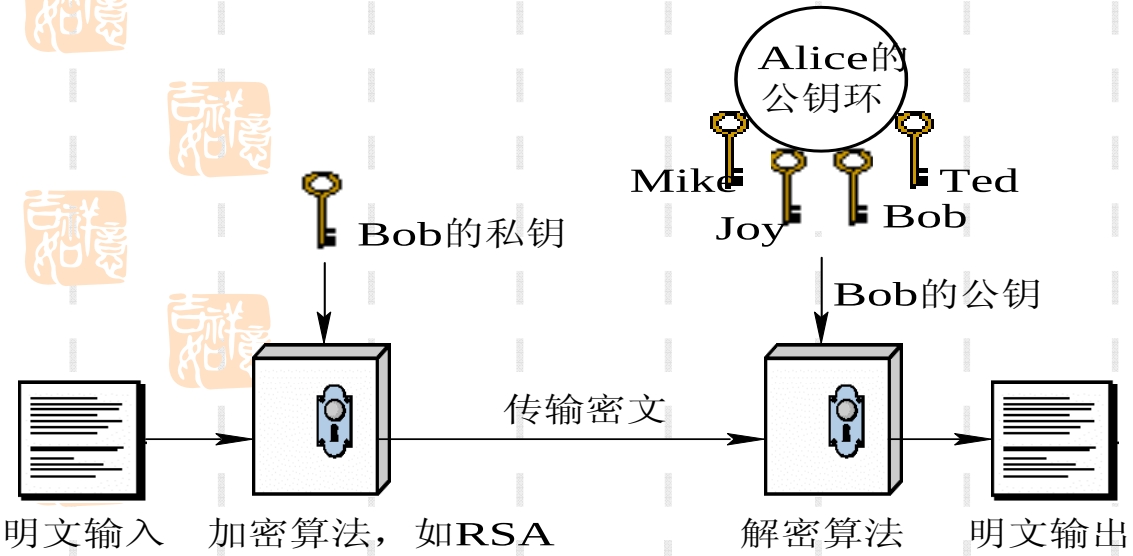
保密与签名的不同



- 通信保密：将接收方的公钥作为加密密钥，接收方用自己的私钥作为解密密钥，通信双方不需要交换密钥就可以实现保密通信。

Bob

Alice



- 数字签名：将发送者私钥作为加密密钥，公钥作为解密密钥，由一个用户对数据加密而使多个用户解读
- 接收方通过发送方对应的公钥来鉴别消息，并且发送方不能对自己的签名进行否认

数字签名方案描述

- 数字签名使用：
  - 对称密码系统
    - 可以对文件进行签名
    - 但此时需要可信任的第三方仲裁
  - 公钥密码系统
    - 公开密钥算法也能用于数字签名。此时，发方用私钥对文件进行加密就可以获得安全的数字签名。
    - 公开密钥算法加密效率较低，因而发送方并不对整个文件签名，而只对文件的散列值签名。

数字签名的执行方式

- 数字签名的执行方式有两类：直接数字签名方式和具有仲裁的数字签名方式

1. 直接方式（对称/不对称）

直接方式是指数字签名的执行过程只有通信双方参与，并假定双方有共享的秘密密钥，或者接收一方知道发方的公开密钥。

直接数字签名有一些共同的缺点：方案的有效性依赖于发送方秘密密钥（加密密钥）的安全性。

2. 具有仲裁的方式（对称/不对称）

具有仲裁的数字签名是在通信双方的基础上引入了第三方仲裁者参与

下面给出几个需要仲裁者的数字签名方案。其中S表示发送方，R表示接收方，A是仲裁者，M是传送的消息。

数字签名的执行方式

2. 具有仲裁的方式

方案一 对称加密，仲裁者可以看到消息内容

该方案的前提是每个用户都有与仲裁者共享的秘密密钥。数字签名过程如下：

(1) S→A：$M||E_{KSA}[ID_S||H(M)]$

(2) A→R：$E_{KAR}[ID_S||M||E_{KSA}[ID_S||H(M)]||T]$

其中E是对称密钥加密算法，K_{SA}和K_{AR}分别是仲裁者A与发送方S、接收方R的共享密钥，H(M)是M的散列值，T是时间戳，ID_S是S的身份标识。

从上面过程中可以看出，A的存在可以解决直接数字签名可能产生的问题。这时仲裁者起着关键的作用，这必须要求：

(1)发送者S必须确信仲裁者A不会泄露K_{SA}，也不会产生虚假的数字签名；

(2)接收方R必须确信仲裁者A只有在散列码正确且发送方S的数字签名被证实的情况下才发送；

(3) 通信双方必须确信仲裁者A能公平的解决争端。

注：该方案中消息以明文方式发送，未提供消息的机密性保护。

数字签名的执行方式

方案二：对称加密，仲裁者不能看到消息内容

该方案的前提是每个用户都有与仲裁者共享的秘密密钥，而且两两用户间也有共享密钥。数字签名过程如下：

(1)S→A：$ID_S||E_{KSR}[M]||E_{KSA}[ID_S||H(E_{KSR}[M])]$

(2)A→R：$E_{KAR}[ID_S||E_{KSR}[M]||E_{KSA}[ID_S||H(E_{KSR}[M])]||T]$

其中K_{SR}是S，R的共享密钥。

本方案虽然对消息M提供了保密性，但是仍存在与方案一相同的问题：

(1) 仲裁者和发送方可以合谋来否认曾经发送过的消息；

(2) 仲裁者和接收方可以合谋来伪造发送方发的签名。

数字签名的执行方式

方案三：公钥加密，仲裁者不能看到消息内容

该方案的前提是每个用户都能安全获取仲裁者和其它用户的公开密钥。数字签名过程如下：

(1) $S \rightarrow A$：$ID_S || E_{KRS}[ID_S || E_{KUR}[E_{KRS}[M]]]$

(2) $A \rightarrow R$：$E_{KRA}[ID_S || E_{KUR}[E_{KRS}[M] || T]$

其中K_{RS}和分别K_{RA}是发送方S和仲裁者A的私钥，K_{UR}是接收方R的公钥

与前两中方案相比，方案三有许多优点：

(1) 在方案执行以前，各方都不必有共享的信息，从而可以防止合谋；

(2) 只要仲裁者的私钥不被泄露，任何人包括发送方就不能重放消息；

(3) 对任何第三方（包括A），S发往R的消息都是保密的。

其他类型的签名

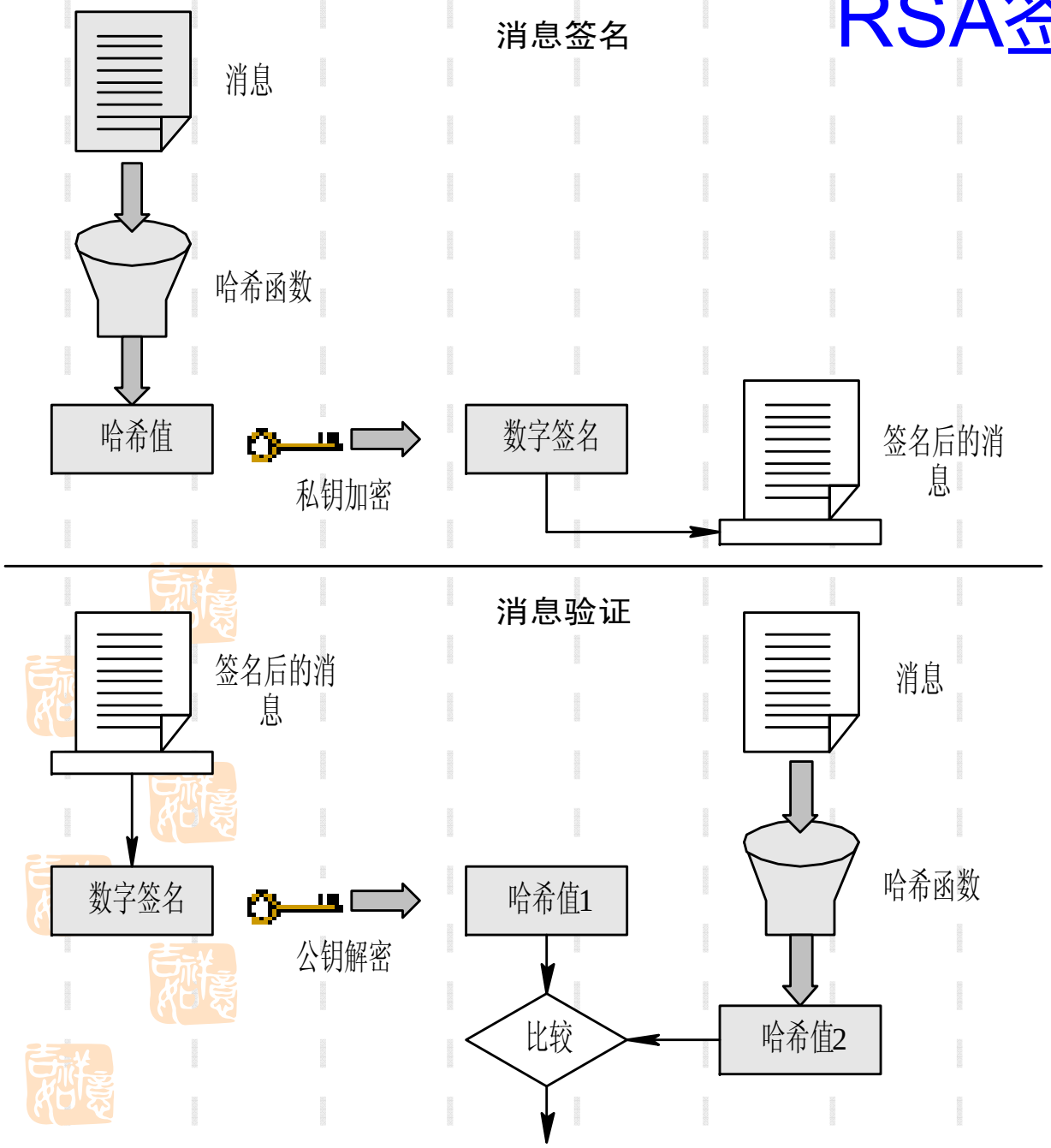
- 盲签名
  - 签名者不知道所签署文件的内容
- 群签名
  - 一组人拥有签名的权利，每个人的签名等效
- 门限签名
  - 一组人中只有达到规定数目人的同意，才能产生签名
- 代理签名
  - 签名人临时将签名权交由代理人行使
- 属性签名
  - 签名的权限随身份(属性)改变而调整
- 多重签名
  - 多个人的会签
- …

数字签名

- 数字签名概述
- 基于公钥密码体制的数字签名方案
- 特殊数字签名方案

- 基于公钥密码体制的数字签名方案
  - RSA数字签名方案
  - ElGamal数字签名方案
  - 数字签名标准DSS
  - 基于椭圆曲线密码的数字签名算法ECDSA

RSA签名基本思想


消息签名
消息
哈希函数
哈希值
私钥加密
数字签名
签名后的消息
消息验证
签名后的消息
数字签名
公钥解密
哈希值1
比较
消息
哈希函数
哈希值2

RSA签名基本思想

- 准备
  - 待签名的消息为M
  - H是一个公开的Hash函数
  - Hash函数的值域为{0,1,…,n-1}
- 签名
  - 1. 取M的Hash值，得到H(M)∈{0,1,…,n-1}
    - 由于Hash函数将任意长度大小的消息映射为固定长度大小的Hash值，所以对签名消息的长度不作限制
  - 2 利用自己的私钥d对H(M)进行签名
    - 得到$S=H(M)^d \pmod n$
- 验证
  - 1. 将消息签名对(M,S)发送给收方
  - 2 收方得到(M,S)后 恢复$V=S^e \pmod n$, 验证V=H(M)?

RSA数字签名方案

RSA数字签名方案

基于RSA公钥密码体制的数字签名方案通常称为RSA数字签名方案。

RSA数字签名体制的基本算法可以表述如下：

(1) 系统初始化过程

① 产生两个大素数 p，q；计算n=p×q；

② 随机选取一个与Φ(n)互素的整数e作为公钥，私钥d满足 $ed=1 \bmod \Phi(n)$；

用户A将公开公钥e与n，而私钥d，以及p，q则严格保密

(2) 签名产生过程

用户A对消息M∈Zn进行签名，计算

$S_A=Sig(M)=M^d \bmod n$ ； 其中*d*为签名方A的私钥。

并将S_A附在消息M后作为对用户对消息M的签名。

实际上可以对M的报文摘要进行签名 $r=sig(m)=(H(M))^d \bmod n$， 其中，

H(M)计算消息M的消息摘要，可由散列函数SHA−1或MD5得到；

r即为对消息的签名

RSA数字签名方案

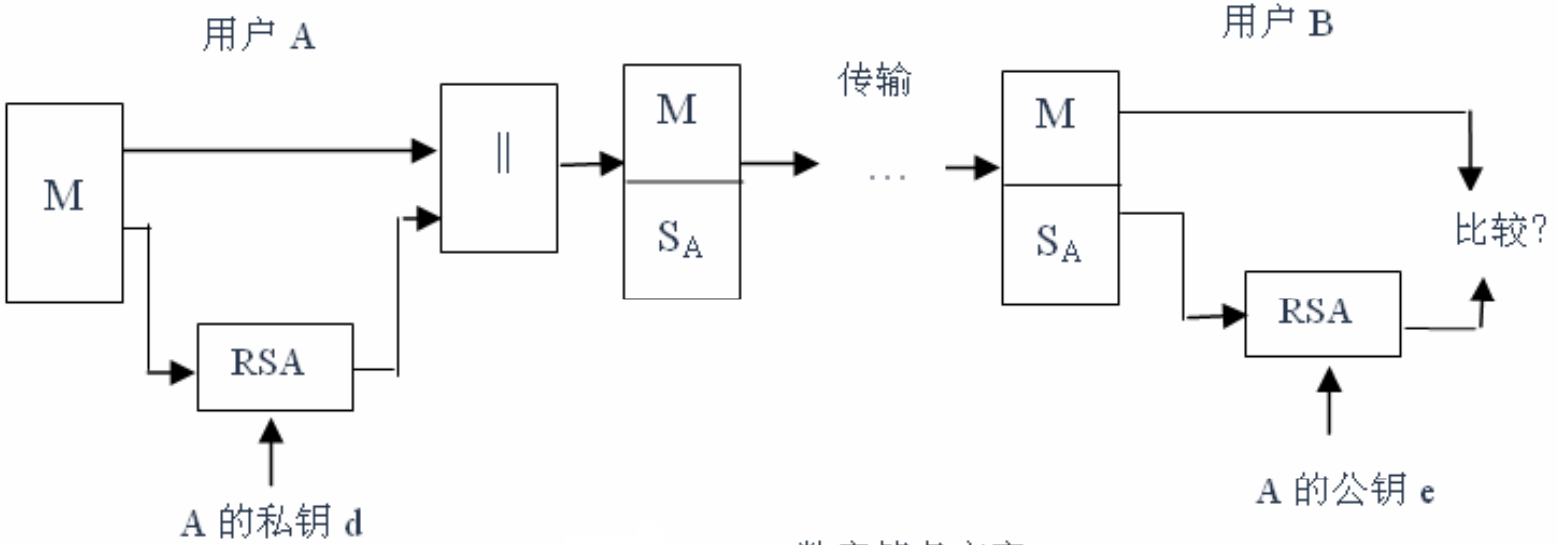
(3) 签名验证过程

假设用户B要验证用户A 对消息M的签名，用户计算

$M' = S_A^e \bmod n$；其中e为签名方A的公钥。

并判断M′ 与M是否相等。如果相等则相信签名确实为A所产生，否则拒绝确认该签名消息。

RSA数字签名算法如下所示：



RSA 数字签名方案

RSA数字签名方案

对于RSA数字签名方案的应用及安全性需要注意以下的几个问题：

(1) 由于公开密钥密码体制一般速度都比较慢，这样当消息比较长时，整个签名与验证过程都会相当的慢。一般对消息M的散列值签名

(2) RSA数字签名算法的安全性取决于RSA公钥密码体制的安全性（基于大整数分解的困难性）。

(3) 如果消息M_1、M_2的签名分别是S_1和S_2，则任何知道M_1，S_1，M_2，S_2的人都可以伪造对消息M_1M_2的签名S_1S_2，这是因为在RSA的数字签名方案中，$Sig(M_1M_2) = Sig(M_1)Sig(M_2)$。

- 基于公钥密码体制的数字签名方案
  - RSA数字签名方案
  - ElGamal数字签名方案
  - 数字签名标准DSS
  - 基于椭圆曲线密码的数字签名算法ECDSA

ElGamal数字签名方案

- ElGamal数字签名方案是T.ElGamal在1985年发表关于ElGamal公开密钥密码时给出的两个方案之一

1. ElGamal数字签名算法描述

(1)系统初始化过程

ElGamal系统初始化过程用来设置系统公共参数和用户的密钥。

① 系统公共参数

a. 选择大素数p，使得Z_p中的离散对数问题为困难问题；

b. 选择乘法群Z_p^*上的一本原元α。

② 用户选择密钥

每个用户选择一随机数x，且$1 \le x < p-1$，计算

$$y = \alpha^x \bmod p$$

用户将x作为自己的私钥（用于签名），将y作为自己的公钥（用于签名验证）。

整个系统公开的参数有大素数p、本原元α以及每个用户的公钥；而每个用户的私钥则严格保密。

ElGamal数字签名方案

(2) 签名过程

给定消息M，签名者A将进行如下计算：

①选择随机数$k \in Z_p^*$，且k与$(p-1)$互素；

②签名方对消息M进行散列压缩得到消息散列码H(M)，并计算:

$r=\alpha^k \bmod p$；

$s=(H(M)-x\cdot r)\,k^{-1} \bmod p$；其中$x$为签名方A的私钥。

③ 用户A将（r，s）作为对消息M的数字签名，与消息M一起发送给接收方

(3) 验证签名过程

接收方B在收到消息M与数字签名(r，s)后：

① 计算消息M的散列码H(M)；

② 计算$y^r r^s \bmod p$和$\alpha^{H(M)} \bmod p$；其中y为签名方A的公钥。

若两式相等，即$y^r r^s = \alpha^{H(M)} \bmod p$则确认($r$，$s$)为有效签名，否则认为签名是伪造的。

ElGamal数字签名方案

2. ElGamal数字签名算法安全性

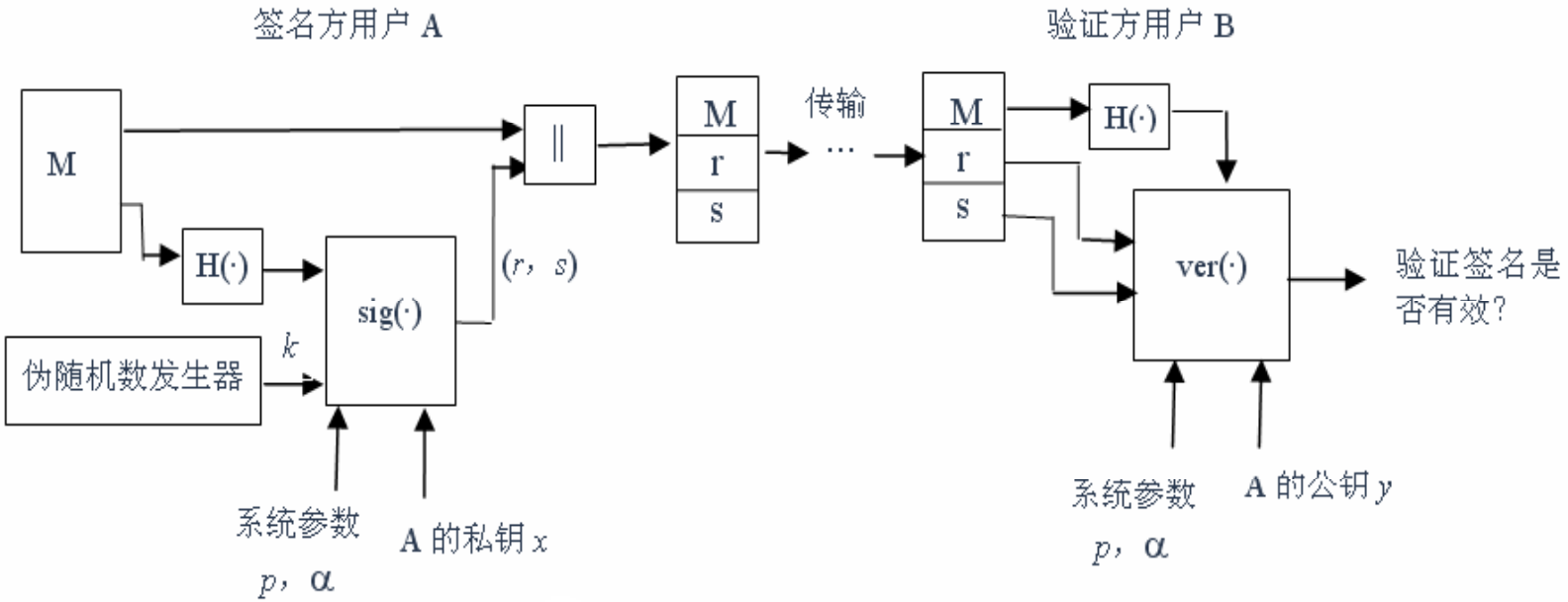
(1) ElGamal数字签名算法是一个非确定性的数字签名算法，对同一个消息M所产生的签名依赖于随机数k

(2) 由于用户的签名密钥x是保密的，攻击者要从公开的验证密钥y得到签名密钥x必须求解有限域上的离散对数问题（$x=\log_\alpha y$）。因此ElGamal数字签名算法的安全性是基于有限域上计算离散对数的困难性。

(3) 在签名时用的随机数k不能被泄露。

(4) 随机数k不能被重复使用。

ElGamal数字签名方案



ElGamal 数字签名方案

- 基于公钥密码体制的数字签名方案
  - RSA数字签名方案
  - ElGamal数字签名方案
  - 数字签名标准DSS
  - 基于椭圆曲线密码的数字签名算法ECDSA

DSA数字签名

- 1991年8月，美国NIST（National Institute of Standard and Technology）公布了数字签名算法DSA，1994年12月1日正式采用为美国联邦信息处理标准

1．DSA数字签名算法描述

(1) DSA的参数

① p为L位长的素数，要求$2^{L-1}<p<2^{L}$，512≤L≤1024，并且L为64的倍数，即位长度在512到1024之间，长度增量为64位。

② q是160位长的素数，且为p−1的因子，$2^{159}<q<2^{160}$

③ $g=h^{(p-1)/q} \bmod p$，其中h是一个整数，满足1<h<p - 1并且$g=h^{(p-1)/q} \bmod p>1$。

④ 随机选择整数x作为用户的私钥，要求0<x<q。

⑤ 计算用户的公钥$y=g^{x} \bmod p$。显然，给定x计算y是容易的，但是给定y求x是离散对数问题。

前三个参数p、q、g是公开的；x为私钥，y为公钥；

随机产生一个k: x和k用于数字签名，必须保密；对于每一次签名都应该产生一次k

数字签名标准DSS

1．DSA数字签名算法描述

(2) 签名过程

假设用户A要对消息M进行数字签名，然后发送给用户B.

① 发送方A秘密选取随机整数k，0<k<q；

② 发送方计算

r = (g^k mod p) mod q；

$s = k^{-1}(H(M) + xr) \bmod q$ ；其中x为签名方A的私钥。

其中H(M)是使用SHA-1生成的消息M的散列码，（r，s）就是消息M的数字签名，k^{-1}是k模q的乘法逆元，为了安全起见，每次签名应当随机选取不同的k。若r=0或s=0则返回①重新计算。

用户A发送消息: M，r，s

数字签名标准DSS

(3)签名验证过程

如果接收者收到消息M，r，s后，首先验证0<r<q，0<s<q，如果通过则计算：

$w=s^{-1} \bmod q$；

$u_1= H(M)w \bmod q$；

$u_2= r\ w \bmod q$;

$v=[(g^{u_1} y^{u_2}) \bmod p] \bmod q$ ；其中y为签名方A的公钥。

如果v=r则确定签名合法，可以认为收到的消息是可信的，否则消息可能被篡改。

定理1 设p和q是素数且满足$q|(p-1)$，h是小于p的正整数，$g=h^{(p-1)/q} \bmod p$，则$g^p \bmod p=1$，并且假如$m=n \bmod q$，则$g^m=g^n \bmod p$

定理2. 如果(r，s)是合法用户A采用DSA数字签名算法对消息M的签名，则一定有v=r成立。

DSA总结

- 对消息m签名：
  - $r = (g^k \bmod p) \bmod q$
  - $s = (k^{-1} (SHA-1(m) + xr)) \bmod q$
    - r和s就是签名
- 验证签名时，计算：
  - $w = s^{-1} \bmod q$
  - $u1 = (SHA-1(m) \times w) \bmod q$
  - $u2 = (rw) \bmod q$
  - $v = ((g^{u1} \times y^{u2}) \bmod p) \bmod q$
  - 如果v=r，则签名有效

DSA数字签名算法的安全性分析

DSA算法也是非确定性的数字签名算法，对消息M它的签名依赖于随机数r，这样相同的消息就可能产生不同的签名。

DSA的安全性也是基于计算有限域离散对数的困难性，鉴于有限域上计算离散对数问题的进展，一般认为512位的DSA算法无法提供较长期的安全性，而1024位的安全性值得信赖。

- 基于公钥密码体制的数字签名方案
  - RSA数字签名方案
  - ElGamal数字签名方案
  - 数字签名标准DSS
  - 基于椭圆曲线密码的数字签名算法ECDSA

基于椭圆曲线密码的数字签名算法ECDSA

设E为定义在有限域Fp上的椭圆曲线（p为大于3的素数）。以E(Fp)记椭圆曲线E在Fp中的有理点集，它是一个有限群。在E(Fp)中选一个阶(order)为n的基点G，通常要求n是一个大素数

每个每户选取一个大整数d(1<d<n)作为签名私钥（严格保密），而以点Q=d·G作为其公钥，这样便形成一个椭圆曲线密码系统（ECC）。该系统的公开参数有：

① 基域Fp；

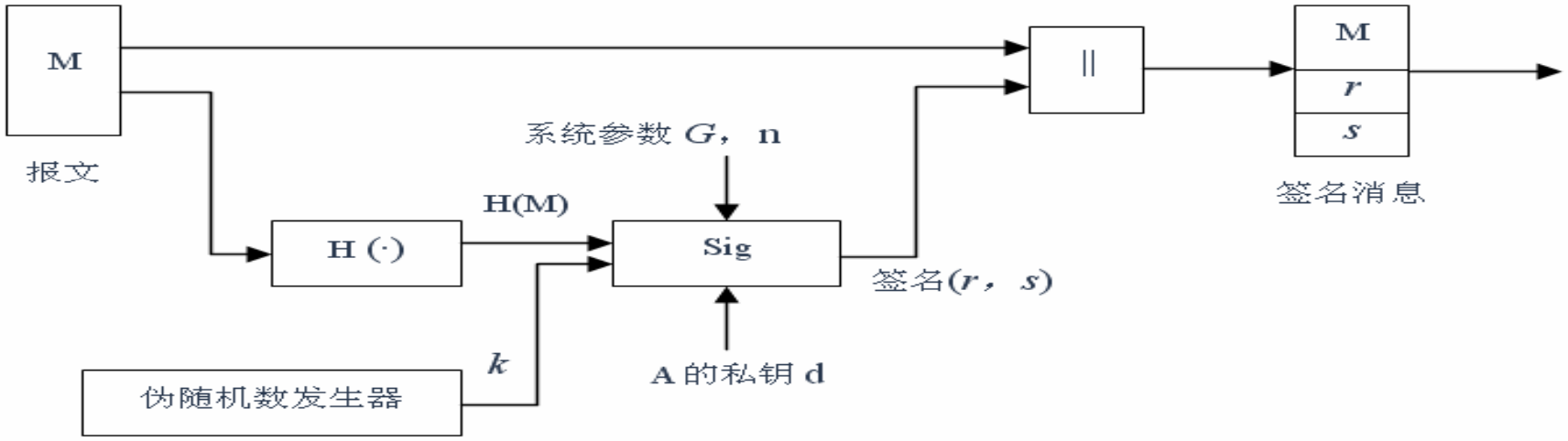
② 椭圆曲线E（如p>3时，由 $y^2 = x^3 + ax + b,\ \ 4a^3 + 27b^2 \neq 0,$ 定义的曲线）；

③ 基点G及其阶n；

④ 每个用户的公钥$Q = d\,G$。

基于椭圆曲线密码的数字签名算法ECDSA

用户A要对报文M进行数字签名，其签名过程如下



ECDSA 的签名过程

签名时，用户A进行以下操作：

① 选择一个随机数k，1<k<n-1;

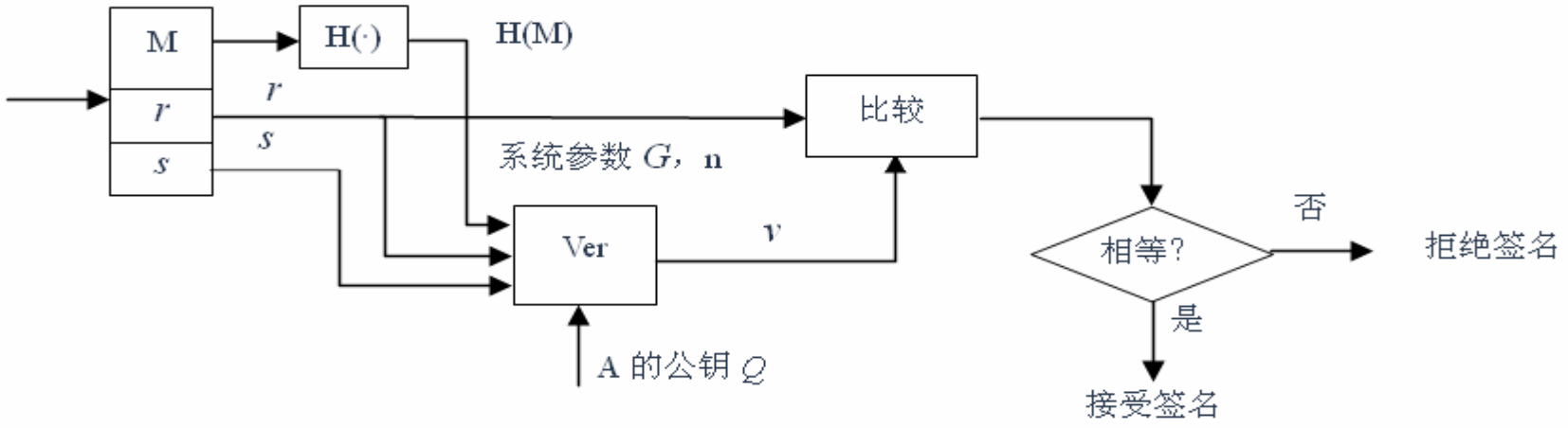
② 计算：$kG = (x_1, y_1)$，取r=x1 mod n 。若r=0，回到①；

③ 计算：$s=k^{-1}(h(M)+d\cdot r) \bmod n$, 其中$d$为签名方A的私钥。

若s=0，回到①否则用户A把（r，s）作为消息M的数字签名，与M一起发送给接收方

基于椭圆曲线密码的数字签名算法ECDSA

接收方B在受到消息M和数字签名（r，s）后，其签名验证过程如下所示：



ECDSA 的验证过程

验证时，接收方B进行以下操作：

① 验证r和s是否是区间[1,n-1]中的整数，若不是则拒绝签名：

② 计算$u_1=h(M)\cdot s^{-1} \bmod n$ 和$u_2=rs^{-1} \bmod n$

③ 利用签名方A的公钥Q计算：$X(x_2, y_2)= u_1.G + u_2.Q$，取$v=x_2 \bmod n$

④ 若v=r，接受签名，否则拒绝签名。

数字签名

- 数字签名概述
- 基于公钥密码体制的数字签名方案
- 特殊数字签名方案

- 特殊数字签名方案
  - 盲签名
  - 群签名

1. 盲数字签名

盲签名基本思想：

在常规的数字签名方案中，签名者总是先知道数据的内容后才实施签名，这是通常的办公事务所需要的。但有时却需要某个人对某数据签名，而又不能让他知道数据的内容，这种签名方式称为盲数字签名（Blind Signature），简称盲签名。

盲签名特点

与普通数字签名比较，盲数字签名具有两个显著特点：

(1) 消息的内容对签名者是不可见的。

(2) 在签名被接收者公开后，签名者不能追踪签名。

盲数字签名

具体实现

下面介绍D.Chaum利用RSA算法设计的第一个盲数字签名方案

设签名者B的公钥为e，私钥为d，模数为n。用户A有消息M需要签名者B对消息M进行盲签名。

(1) 用户A随机选取一个整数$1 \le k \le M$，做盲化处理：

$$t = M \cdot k^e \bmod n$$

将t发送给签名者B;

(2) 签名B对t进行签名 $s = t^d = (M \cdot k^e)^d \bmod n = M^d \cdot k \bmod n$

然后将签名s发送给用户A；

(3) 用户A收到盲签名信息s后，通过解盲计算得到B对消息M的签名信息S：

$$S = s \cdot k^{-1} \bmod n = M^d \cdot k \cdot k^{-1} \bmod n = M^d \bmod n$$

至此，用户A得到了签名者B对消息M的盲签名。

2. 群签名

基本思想

某个群组中的任何一个成员都可以以群组的名义匿名签发消息，验证者可以确认数字签名来自该群组，但不能确认是群组中的哪一个成员进行的签名。但当出现争议时，借助于一个可信的机构或群成员的联合就能识别出那个签名者。满足以上要求的数字签名就是群签名（Group signature）。

一个群签名方案主要由签名算法、验证算法和识别算法组成。

基本特征

一般来说，群签名方案涉及的实体包括群组、群组成员（签名者）、签名验证者及一个可信机构，并具有以下特点：

（1）只有群组的成员能代表这个群组对消息签名，并产生群签名；

（2）签名验证者能验证这个签名是该群组的一个有效签名，但不能辨认是群组中哪个成员产生的签名。

（3）在后来发生争端的情况下，通过可信机构能识别出那个签名者。

群签名

基本性质

一个安全的群签名算法必须具有以下性质：

（1）匿名性：给定某个消息的一个有效群签名后，除了可信机构之外，任何人识别出签名人的身份在计算上是不可行的。

（2）不关联性。除了可信机构之外，确定两个不同的有效群签名是否源自同一个群成员所签，在计算上是不可行的。

（3）防伪造性。只有群成员才能代表这个群组产生有效的群签名。

（4）可跟踪性。可信机构在必要时可以识别出群签名者的身份。

（5）可开脱性。可信机构及任何群成员都不能以其他群成员的名义产生出有效的群签名，这样群成员不必为他人产生的群签名负责。

（6）抗合谋攻击。即使一些群成员合谋也不能产生一个有效的群签名，使得可信机构不能将群签名与其中一个群成员的身份联系起来。

The End!