密 码 学

第十二讲　数字签名基础

王后珍

武汉大学国家网络安全学院

空天信息安全与可信计算教育部重点实验室

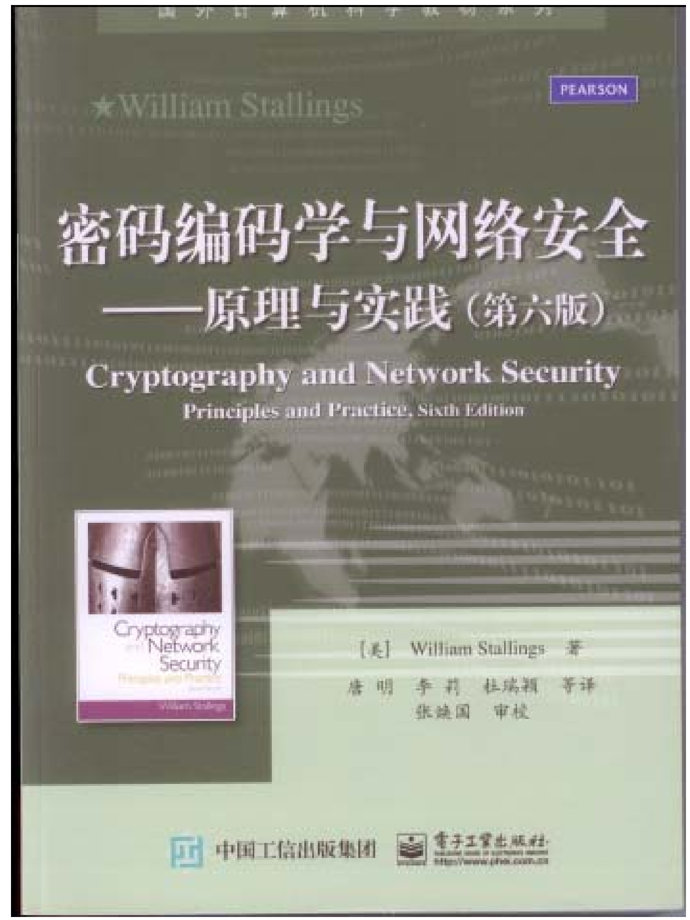
目 录

目 录

教材与主要参考书

教材

参考书



本讲内容

一、数字签名的基本概念

二、数字签名模型

三、利用RSA密码实现数字签名

四、利用EIGamal密码实现数字签名

一、数字签名的基本概念

①在人们的工作和生活中，许多事物的处理需要当事者签名。

②签名起到确认、核准、生效和负责任等多种作用。

③签名是证明当事者的身份和数据真实性的一种信息。

④签名可以用不同的形式来表示。

一、数字签名的基本概念

⑤在传统的以书面文件为基础的事物处理中，采用书面签名的形式：

手签、印章、手印等

⑥书面签名得到司法部门的支持。

⑦在以计算机文件为基础的现代事物处理中，应采用电子数字形式的签名，即数字签名（**Digital Signature**）。

⑧数字签名已得到中国和其它一些国家的法律支持。

一、数字签名的基本概念

⑨一种完善的签名应满足以下三个条件：

- 签名者事后不能抵赖自己的签名；
- 任何其他人不能伪造签名；
- 如果当事的双方关于签名的真伪发生争执，能够在公正的仲裁者面前通过验证确认其真伪。

一、数字签名的基本概念

⑩数字签名基于密码技术，其形式是多种多样的：

通用签名、仲裁签名、盲签名、群签名、门限签名，代理签名等。

- 1994年月美国政府正式颁布了美国数字签名标准DSS（Digital Signature Standard）。
- 1995年我国也制定了自己的数字签名标准（GB15851－1995）。
- 2004年我国颁布了《中华人民共和国电子签名法》。

二、数字签名模型

- 一个数字签名体制包括两个方面的处理：
  - 施加签名：为数据产生签名
  - 验证签名：验证签名的真伪
- 设施加签名的算法为*SIG*，产生签名的密钥为*K*，被签名的数据为*M*，产生的签名信息为*S*，则有

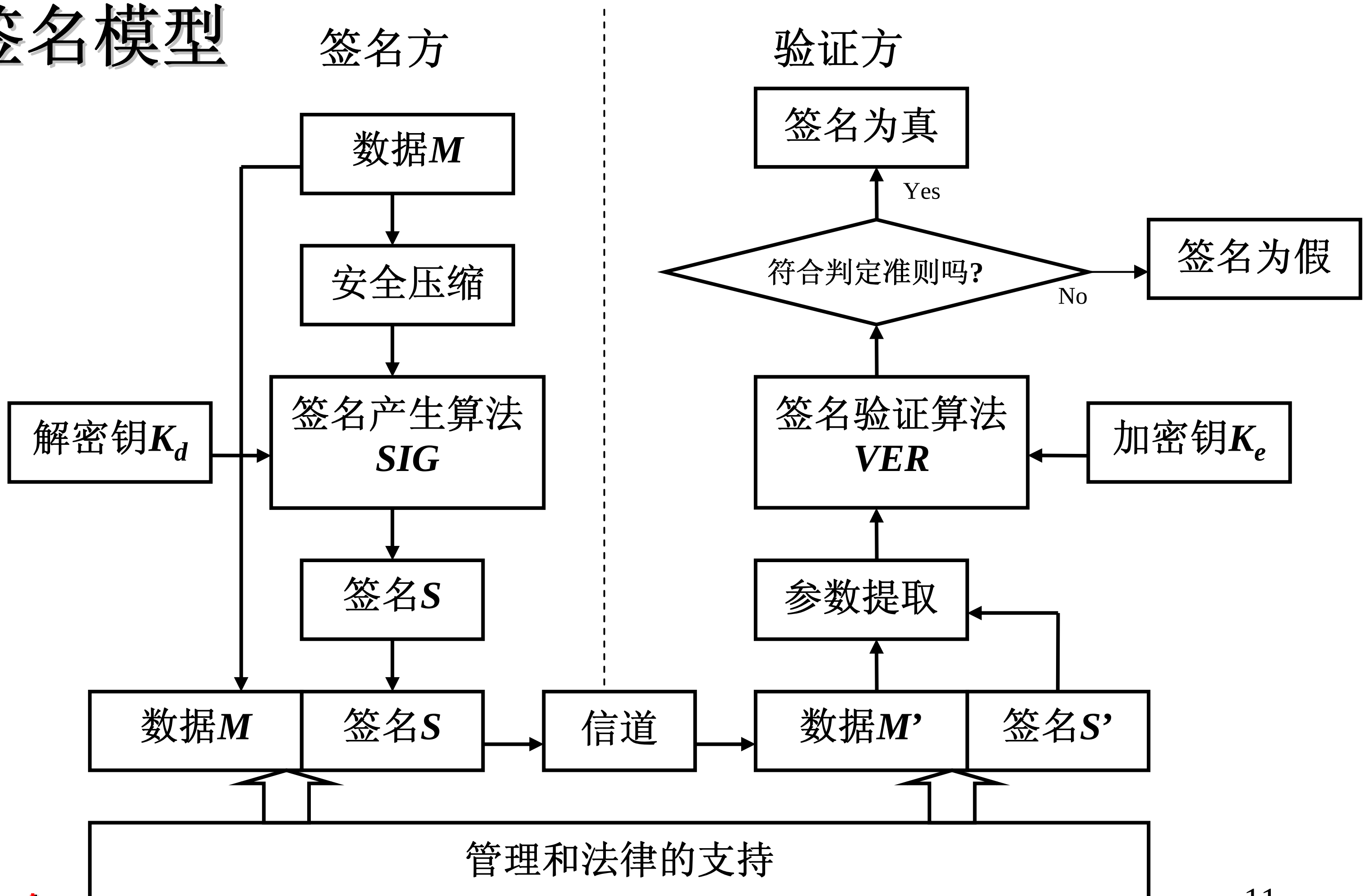
$$S = SIG(M, K)$$

- 设验证签名的算法为 *VER* ，用*VER*对签名S进行验证，可鉴别 S的真假。即

$$VER(S, K) = \begin{cases} \text{真，当} S = SIG(M, K) \\ \text{假，当} S \neq SIG(M, K) \end{cases}$$

二、数字签名模型

- 数字签名模型



武漢大學

二、数字签名模型

- 签名函数必须满足以下条件：

① 当$M' \neq M$时，有$SIG(M', K) \neq SIG(M, K)$。

- 条件①要求签名*S*至少和被签名的数据*M*一样长。当*M*太长时，应用很不方便。
- 将条件①改为：虽然当$M' \neq M$时，存在$S = S'$，但对于给定的*M*或*S*，要找出相应的M'在计算上是不可能的。

② 签名 *S* 只能由签名者产生，否则别人便可伪造，于是签名者也就可以抵赖。

- *SIG*使用签名者自己的解密钥K_d，K_d只有签名者一人拥有，所以别人不能产生。同理，别人也不能伪造签名。

武汉大学

二、数字签名模型

③ 收信者可以验证签名*S*的真伪。这使得当签名*S*为假时收信者不致上当。

- ***VER***使用签名者的公开加密钥K_e，收信者和第三方都可公开验证签名*S*的真伪。从而确保签名*S*为假时收信者不上当，当签名*S*为真时可阻止签名者的抵赖。

④ 签名者也应有办法鉴别收信者所出示的签名是否是自己的签名。这就给签名者以自卫的能力。

- 除了与③一样的理由外，还有管理与法律的支持，所以可以通过公开验证签名的真伪解决纠纷。

二、数字签名模型

- 凡是能够构成安全*SIG*和*VER*的公钥密码都可实现数字签名
  - 数字签名并不要求*SIG*与*VER*之间具有互逆关系：

    $$VER（S，K_e）= VER（SIG(M，K_d)，K_e）= M$$

  - 也不要求*SIG*与*VER*之间具有可交换性

    $$VER（SIG(M，K_d)，K_e）$$

    $$= SIG（VER(M，K_e)，K_d）= M$$

  - 如果一个公钥密码体制能够满足上面两式，那将更好。例如**RSA**密码。难怪学术界把**RSA**密码称为风格优雅的密码。

三、利用RSA密码实现数字签名

1、利用公钥密码实现数字签名的一般方法

- 凡是能够构成安全*SIG*和*VER*的公钥密码都可实现数字签名
  - RSA密码
  - ElGamal密码
  - 椭圆曲线密码
  - 许多其他密码

三、利用RSA密码实现数字签名

1、利用公钥密码实现数字签名的一般方法

- 为了实施数字签名，应成立管理机构：
  - 制定规章制度，
  - 统一技术标准，
  - 用户登记注册，
  - 纠纷的仲裁，
  - 其它。

武漢大學

三、利用RSA密码实现数字签名

2、利用RSA密码实现数字签名：

- 对于RSA密码
  - $D(E(M))=(M^e)^d=M^{ed}=(M^d)^e=E(D(M)) \bmod n$，
    所以RSA可同时确保数据的秘密性和真实性。
  - 在这里，*SIG=D*，*VER=E*

- 因此利用RSA密码可以同时实现数据加密和数字签名。

三、利用RSA密码实现数字签名

2、利用RSA密码实现数字签名：

(1)、签名算法

- 设M为明文，$K_{eA}=<e,n>$是A的公开加密钥，
  $K_{dA}=<d,p,q,\phi(n)>$是A的保密的解密钥，

  则A对M的签名过程是，

  $$S_A = D(M,K_{dA}) = (M^d) \bmod n$$

  S_A便是A对M的签名。
- 验证签名的过程是，

  $$E(S_A, K_{eA}) = (M^d)^e \bmod n = M$$
- 如果收信者验证得到正确的数据M，则签名为真。

三、利用公钥密码实现数字签名

- 上述验证中，如果收信者验证得到正确的数据M，则判定签名为真。有时收信者事前不知道M，如何判定？

①方法一：合理设计明文的数据格式：

发方标识	收方标识	报文序号	时间	数据	纠错码

$$M = < \mathrm{A},\ \mathrm{B},\ I,\ T,\ \mathrm{DATA},\ \mathrm{CRC} >$$

记 $H = < \mathrm{A},\ \mathrm{B},\ I,\ T >$ 。

- A以 $< H,\ D(M,K_{dA}) >$ 为最终报文发给B，其中H为明文形式。

三、利用公钥密码实现数字签名

- 只要用A的公钥验证签名并恢复出正确的附加信息 H = < A，B，I，T >，便可断定明文M是否正确。

  记接收到的附加信息为H，恢复出的为H'，仅当H=H'时判定签名为真。

- 设附加信息 H=<A，B，I，T> 的二进制长度为L，则错判概率

$$p_e \leqslant 2^{-L}。$$

三、利用公钥密码实现数字签名

②方法二：对Hash（*M*）签名

- 签名改为：对Hash（*M*）签名，而不直接对*M*签名。

数据 *M*	Hash(*M*)

- 签名：$S = D(\text{Hash}(M), K_{dA})$
- 传输格式：< *M*，*S* >

数据 *M*	签名*S*

- 设收到的数据为< *M'*，*S'*>，仅当$\text{Hash}(M')=E(S',K_{eA})$时，判定*M*是正确的，签名*S*是正确的。

三、利用RSA密码实现数字签名

(2)、对RSA数字签名的攻击

①一般攻击:

- 因为e和n是用户A的公开密钥，所以任何人都可以获得并使用e和n。攻击者可随意选择一个数据Y，并用A的公钥计算

$$X = (Y)^{e} \bmod n$$

- 因为 $Y = (X)^{d} \bmod n$ ，于是可以用Y 伪造A的签名。因为Y 是A对X 的一个有效签名。
- 注意：这样的 X 往往无正确语义！因此，这种攻击在实际上有效性不大！

三、利用RSA密码实现数字签名

⑵、对RSA数字签名的攻击

②利用已有的签名进行攻击:

- 攻击者选择随机数据 M_3，且$M_3=M_1M_2 \bmod n$ 。
- 攻击者设法让A对M_1和M_2签名：

  $S_1=(M_1)^d \bmod n$， $S_2=(M_2)^d \bmod n$
- 于是可以由S_1和S_2计算出A对M_3的签名。因为

$S_1S_2=(M_1)^d(M_2)^d \bmod n=(M_3)^d \bmod n=S_3$

- 对策：**A不直接对数据*M* 签名，而是对HASH(*M*)签名。**

武漢大學

三、利用RSA密码实现数字签名

⑵、对RSA数字签名的攻击

②利用已有的签名进行攻击:

- 此时：

$S_1 = (\mathrm{HASH}(M_1))^d \bmod n$， $S_2 = (\mathrm{HASH}(M_2))^d \bmod n$

而，

$(\mathrm{HASH}(M_1))^d (\mathrm{HASH}(M_2))^d \neq (\mathrm{HASH}(M_1 M_2))^d \bmod n$

- 所以：$S_3 \neq S_1 S_2$
- 于是不能由S_1和S_2计算出A对M_3的签名。

三、利用RSA密码实现数字签名

(2)、对RSA数字签名的攻击

③攻击签名获得明文:

- 攻击者截获C，$C=(M)^e \bmod n$ 。
- 攻击者选择小的随机数r，计算:

$$x = r^e \bmod n，y = xC \bmod n ， t = r^{-1} \bmod n$$

- 攻击者让A对y签名，

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

于是攻击者又可截获S

- 攻击者计算$tS=r^{-1}y^d=r^{-1}x^dC^d=C^d=M \bmod n$
- 对策：A不直接对数据M签名，而是对HASH(M)签名。

三、利用RSA密码实现数字签名

(2)、对RSA数字签名的攻击

- 结论:
  - 不直接对数据*M*签名，而是对**HASH(*M*)**签名。
  - 使用时间戳
  - 对于同时确保秘密性和真实性的通信，应当先签名后加密。

三、利用RSA密码实现数字签名

(3)、RSA数字签名的应用

- RSA签名已经得到广泛应用
  - 电子商务，可信计算，等
- 举例：PGP
  - 数据*M*经MD5处理，得到MD5（*M*）
  - 利用RSA对HASH(*M*)签名,得到*M*的签名*S*
  - 使用ZIP对< *M*, *S* >压缩
  - 再用IDEA对压缩数据加密： IDEA(ZIP(*M*, *S*))
  - 用RSA对IDEA的密钥加密：RSA(*k*)
  - 形成数据：< IDEA(ZIP(*M*,*S*)), RSA(*k*)>
  - 将数据转换成ASCII码。

四、利用ElGamal密码实现数字签名

- 利用ELGamal密码可以构建安全的*SIG*和*VER*，所以可以实现数字签名

(1)密钥选择

- 选*P*是一个大素数，*p*-1有大素数因子，α是一个模*p*的本原元，将*p*和α公开作为密码基础参数。
- 用户随机地选择一个整数*x*，以*x*作为自己的秘密的解密钥，$1< x < p-1$。
- 计算 $y = \alpha^x \bmod p$，取*y*为自己的公开的加密钥。

四、利用EIGamal密码实现数字签名

(2) 产生签名

设明文为m，$0 \leqslant m \leqslant p-1$, 签名过程如下：

① 用户A随机地选择一个整数 k，$1<k<p-1$，且（k，p-1）=1;

① 计算$r = \alpha^{k} \bmod p$

② 计算$s = (m - xr)k^{-1} \bmod p-1$

③ 取（r，s）作为m的签名,并以$<m, r, s>$的形式发给用户B。

m	r	s

武漢大學

四、利用EIGamal密码实现数字签名

(3) 验证签名

- 用户B接收：$<m, r, s>$
- 用户B用A的公钥y验证：$\alpha^{m}=y^{r}r^{s} \bmod p$是否成立，若成立则签名为真，否则签名为假。
- 签名的可验证性证明如下：

因为$s=(m-xr)k^{-1} \bmod p-1$，

所以 $m=xr+ks \bmod p-1$，

故 $\alpha^{m}=\alpha^{xr+ks}=(\alpha^{x})^{r}(\alpha^{k})^{s}=y^{r}r^{s} \bmod p$，故签名可验证。

四、利用EIGamal密码实现数字签名

(3) 验证签名

- 安全性
  - 从公开密钥 $y = \alpha^x \bmod p$ 求私钥 x是离散对数问题，这是困难的。
  - p-1要有大素数因子,否则易受攻击。
  - 为了安全，随机数k应当是一次性的。否则时间一长，k将可能泄露。因为，

$$x=(m-ks)r^{-1} \bmod p-1,$$

如果知道了m，便可求出保密的解密钥。

四、利用ElGamal密码实现数字签名

(3) 验证签名

- 安全性
  - 如果k重复使用，如用k签名m_1和m_2。于是，

$$m_1 = xr + ks_1 \bmod p\text{-}1,$$

$$m_2 = xr + ks_2 \bmod p\text{-}1,$$

于是，$(s_1\text{-}s_2)k=(m_1\text{-}m_2) \bmod p\text{-}1$

如果知道了m_1和m_2，便可求出k，进而求出保密的解密钥。

  - 由此可知，不要随便给别人签名。
  - 不要直接对m签名，而是对HASH(m)签名。

四、利用ElGamal密码实现数字签名

(4)、**ELGamal**密码签名的应用

- 安全，方便。
- 签名时需要使用随机数k，所以需要有良好的随机数产生器。
- 缺点：由于取（r，s）作为m的签名，所以数字签名的长度是明文的两倍，数据扩张一倍。
- 美国数字签名标准（**DSS**）的签名算法**DSA**是它的一种变形。
- 俄罗斯数字签名标准（**GOST**）也是采用一种**ELGamal**密码签名变种。

SM2签名方案

一、利用椭圆曲线密码实现数字签名

- 利用素域$GF(p)$上的椭圆曲线和$GF(2^m)$域上的椭圆曲线都可以构成椭圆曲密码线签名方案
- 这里只介绍素域$GF(p)$上的椭圆曲线密码签名方案
- 一个椭圆曲线密码由下面的六元组描述：
  - $T=\langle p,a,b,G,n,h\rangle$
  - 其中，p为大于3的素数，p确定了有限域GF（p）；元素$a,b\in GF(p)$，a和b确定了椭圆曲线；G为循环子群E_1的生成元，n为素数且为生成元G的阶，G和n确定了循环子群E_1。h为余因子，$h=|E|/n$ 。
  - $y^2=x^3+ax+b \bmod p$

一、利用椭圆曲线密码实现数字签名

椭圆曲线密码数字签名

(1) 密钥选择

- $y^2=x^3+ax+b \bmod p$
  全体解点和无穷远点构成群，G为其循环子群E_1的生成元，n为素数且为G的阶。
- 用户的私钥：
  随机数$d \in\{1,2,\cdots,n-1\}$
- 用户的公开钥：
  Q点，$Q=dG$
  由Q求d，要求解椭圆曲线离散对数。

ELGamal密码数字签名

(1)密钥选择

- 选P是一个大素数，p-1有大素数因子，α是一个模p的本原元，将p和α公开作为密码基础参数。
- 用户的私钥：
  随机数x，$1<x<p-1$。
- 用户的公钥：
  $y=\alpha^x \bmod p$
  由y求x，要求解离散对数。

一、利用椭圆曲线密码实现数字签名

(2)产生签名（*SIG*）

设明文为m，$0 \leqslant m \leqslant n-1$

① 选择一个随机数k，$k \in \{1,2,\cdots,n-1\}$；

② 计算点$R(x_R, y_R) = kG$，并记 $r = x_R$；

③ 利用私钥d 计算：

$s = (m - dr)k^{-1} \bmod n$；

④ 以<r,s>作为m的签名，并以<m，r，s>的形式发给接收方。

m	r	s

(2) 产生签名

设明文为m，$0 \leqslant m \leqslant p-1$

① 随机地选择一个整数 k，$1<k<p-1$，且$(k，p-1)=1$；

② 计算$r = \alpha^k \bmod p$

③ 利用私钥x计算:

$s = (m - xr)k^{-1} \bmod p-1$

④ 取（r，s）作为m的签名，并以<m，r，s>的形式发给接收方。

m	r	s

一、利用椭圆曲线密码实现数字签名

(3) 验证签名（*VER*）

① 计算$s^{-1} \bmod n$；

② 利用公密钥 Q 计算：

$U(x_U, y_U) = s^{-1}(mG-rQ)$；

③ 如果$x_U = r$，则签名<r,s>为真，否则签名为假。

证明：因为$s=(m-dr)k^{-1} \bmod n$，故，$s^{-1}=(m-dr)^{-1}k \bmod n$，

$U(x_U, y_U) = (m-dr)^{-1}k(mG-rQ)$

$= (m-dr)^{-1}(mkG-krdG)$

$= (m-dr)^{-1}(mR-rdR)$

$= (m-dr)^{-1}R(m-dr) = R(x_R, y_R)$

所以$x_U = x_R = r$.

(3) 验证签名

- 用户B用A的公钥y验证：

  $\alpha^m = y^r r^s \bmod p$，

  若成立则签名为真，否则签名为假。

- 可验证性证明如下：

  因为$s=(m-xr)k^{-1} \bmod p-1$，

  所以 $m=xr+ks \bmod p-1$，

  故 $\alpha^m = \alpha^{xr+ks} = (\alpha^x)^r(\alpha^k)^s = y^r r^s \bmod p$，故签名可验证。

武汉大学

一、利用椭圆曲线密码实现数字签名

(4) 椭圆曲线密码签名的应用

- 安全，密钥短、软硬件实现节省等特点。
- **2000**年美国政府已将椭圆曲线密码引入数字签名标准**DSS**。
- 我国也颁布了椭圆曲线密码签名标准**SM2**。

二、中国商用公钥密码SM2签名算法

1、 推荐使用256位素域$GF(p)$上的椭圆曲线：

$$y^2 = x^3 + ax + b$$

曲线参数：

p = 8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

a = 787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

b = 63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 6E12D1DA

n = 8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

h=1

Gx = 421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

Gy = 0680512B CBB42C07 D47349D2 153B70C4 E5D7FDFC BFA36EA1 A85841B9 E46E09A2

2、密钥：

- 私钥是随机数：d，$d \in [1，n\text{-}1]$
- 公钥是点：$P = dG$

二、中国商用公钥密码SM2签名算法

3、产生签名的算法（SIG）

- 设A发签名消息给B。
- 设待签名消息为M，ID_A是A的标识符，$ENTL_A$是ID_A的长度，d_A是A的私钥，基点$G=(x_G, y_G)$，A的公钥 $P_A=d_AG=(x_A, y_A)$。

  $Z_A=\text{Hash}(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$，
- 这里，Hash=SM3

① 置$M^*=Z_A \| M$；

② 计算$e=\text{Hash}(M^*)$；

③ 用随机数发生器产生随机数$k\in[1, n-1]$；

④ 计算椭圆曲线点$G_1(x_1, y_1)=kG$；

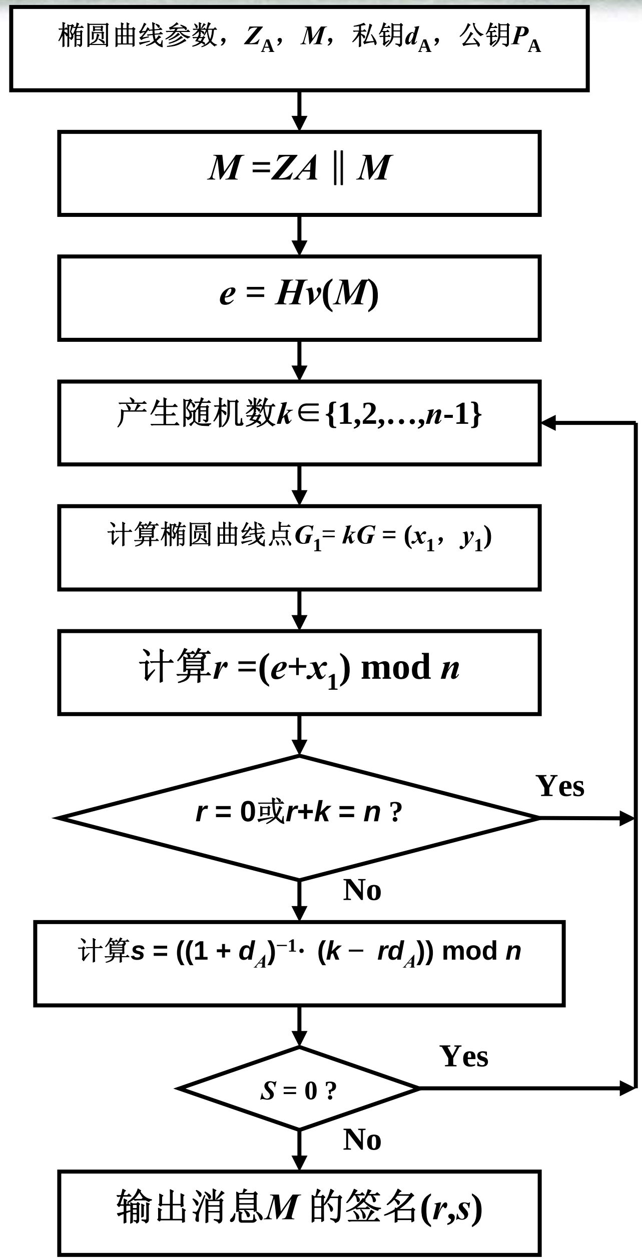
⑤ 计算$r=(e+x_1) \bmod n$，若$r=0$或$r+k=n$则返回③；

⑥ 计算$s=((1+d_A)^{-1}\cdot(k-r\cdot d_A)) \bmod n$，若$s=0$则返回③；

⑦ 以(r,s)作为对消息M的签名。

武汉大学

产生签名算法框图


椭圆曲线参数，Z_A，M，私钥d_A，公钥P_A
$\bar{M} = Z_A \parallel M$
$e = H_v(\bar{M})$
产生随机数$k \in \{1,2,\ldots,n-1\}$
计算椭圆曲线点$G_1 = kG = (x_1, y_1)$
计算$r = (e + x_1) \bmod n$
$r = 0$或$r + k = n$?
Yes
No
计算$s = ((1 + d_A)^{-1} \cdot (k - rd_A)) \bmod n$
$S = 0$?
Yes
No
输出消息M的签名(r,s)

二、中国商用公钥密码SM2签名算法

- 比较SM2签名算法与传统签名算法

① 传统椭圆曲线密码签名算法是原理性的算法，而SM2是实用性的标准算法

② 两者的基本思想一致：

- 都是以 r, s 为签名
- 以 kG 产生 r
- 以 d, r, k 产生 s

二、中国商用公钥密码SM2签名算法

③两者有许多不同

- 传统椭圆曲线签名直接使用m产生签名；
- 而SM2使用$M^* = Z_A \| M$， $e = \mathrm{Hash}(M^*)$
- SM2使用了用户参数和系统参数，起到一定的认证作用，提高了安全性：
  - ID_A是A的标识。$ENTL_A$是ID_A的长度。基点是$G = (x_G, y_G)$
  - A的私钥是d_A，A的公钥是$P_A = d_A G = (x_A, y_A)$
  - $Z_A = \mathrm{Hash}\,(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A)$
- 传统椭圆曲线签名算法计算：点R（x_R, y_R）$= kG$，并记$r = x_R$；
- SM2计算：点$G_1(x_1, y_1) = kG$，且计算$r = (e + x_1) \bmod n$；

二、中国商用公钥密码SM2签名算法

③ 两者有许多不同

- 传统椭圆曲线签名算法计算：

  $s=(m-dr)k^{-1} \bmod n$ ；

- SM2计算：$s = ((1 + d_A)^{-1} \cdot (k - r \cdot d_A)) \bmod n$。$M$ 没有直接出现，而是通过r参与其中；私钥d_A作用了两次。
- SM2增加了合理性检查，确保签名正确，提高安全性。
- 例如第⑤中检查$r+k=n$是否等于n。

如果$r+k=n$，则$k = -r \bmod n$，会使$s = ((1 + d_A)^{-1} \cdot (k - r \cdot d_A)) \bmod n = ((1 + d_A)^{-1} \cdot (-r)(1 + d_A)) = -r \bmod n$ 。 $s = -r \bmod n$，显然是不合适的。

武漢大學

二、中国商用公钥密码SM2签名算法

4、验证签名的算法（VER）

- 设B接收到的消息为M'，签名为(r', s')，P_A为A的公钥。
- 这里，Hash=SM3

① 检验$r' \in [1, n-1]$是否成立，若不成立则验证不通过；

② 检验$s' \in [1, n-1]$是否成立，若不成立则验证不通过；

③ 置$M^* = Z_A \| M'$；

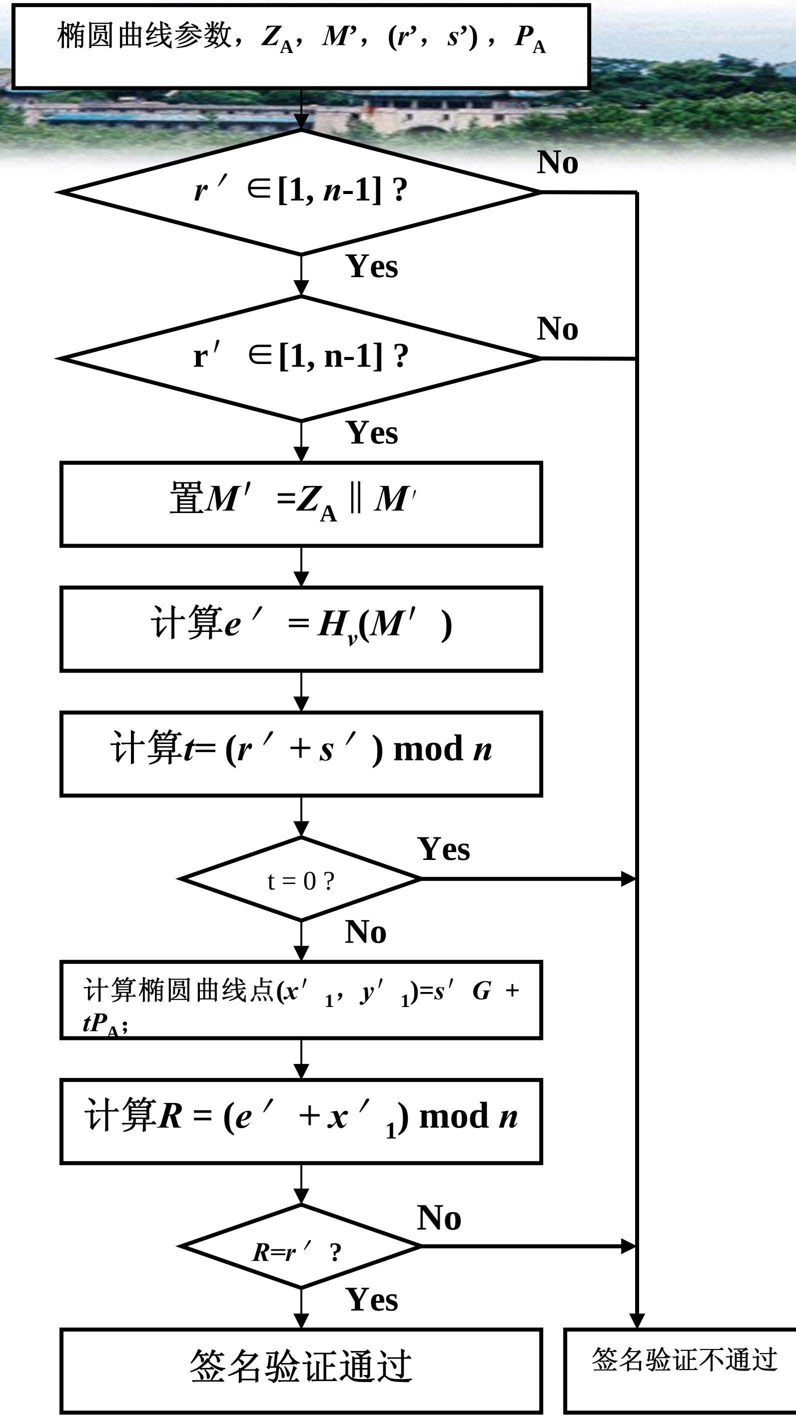
④ 计算$e' = \mathrm{Hash}(M^*)$；

⑤ 计算$t = (r' + s') \bmod n$，若$t = 0$，则验证不通过；

⑥ 计算椭圆曲线点$G_1'(x_1'; y_1') = s'G + tP_A$；

⑦ 计算$R = (e' + x_1') \bmod n$，检验$R = r'$是否成立，若成立则验证通过；否则验证不通过。

验证签名算法框图


椭圆曲线参数，Z_A，M'，(r', s')，P_A
$r' \in [1, n-1]$?
No
Yes
$r' \in [1, n-1]$?
No
Yes
置$\bar{M}' = Z_A \| M'$
计算$e' = H_v(\bar{M}')$
计算$t = (r' + s') \bmod n$
$t = 0$?
Yes
No
计算椭圆曲线点$(x'_1, y'_1) = s'G + tP_A$;
计算$R = (e' + x'_1) \bmod n$
$R = r'$?
No
Yes
签名验证通过
签名验证不通过

二、中国商用公钥密码SM2签名算法

5、验证的正确性

① 因为产生签名算法的第⑤和第⑥步都是mod n运算，且要求$r\neq0$且$s\neq0$，这样就确保了$r\in[1, n-1]$且$s\in[1, n-1]$。如果签名没有被篡改和错误，则必有$r'=r\in[1, n-1]$且$s'=s\in[1, n-1]$。对此进行检验，可发现签名(r,s)是否被篡改或有错误，确保其完整性。这说明验证签名算法①和②的验证是合理的。

二、中国商用公钥密码SM2签名算法

5、验证的正确性

② 签名时确保了$r\neq0$ 且$s\neq0$，如果$t = r + s = 0 \bmod n$，则$r+s$是n的整数倍。但是，由于$r\in[1, n-1]$且$k\in[1, n-1]$，所以$2\leqslant r+k\leqslant 2n-2$。又由于签名时确保了$r+k\neq n$，所以$r+k$不是$n$的整数倍。据签名算法⑥有，$s=\dfrac{k-rd}{1+d}$

所以$r+s=r+\dfrac{k-rd}{1+d}=\dfrac{r+k}{1+d}$，于是 $r+s=\dfrac{r+k}{1+d}$ 也不是n的整数倍。

否则，因d和$1+d$都是正整数，这导致$(r+k)$是n的整数倍，与前面$r+k$不是n的整数倍矛盾。$r+s$不是n的整数倍，即 $r + s \bmod n\neq0$。这说明，如果r'和s'没有被篡改或错误，则有$r'=r$和$s'=s$，则有$t = (r' + s') =(r+ s) \bmod n\neq0$。这说明验证签名算法⑤的验证是合理的。

二、中国商用公钥密码SM2签名算法

5、验证的正确性

③ 可验证性的证明：

- 一方面，$sG + tP_A = sG + (r+s)(dG) = (s+rd+sd)\,G$。
- 另一方面，因为 $s = \dfrac{k - rd}{1 + d}$

  ，故有 $(s + rd + sd) = s(1 + d) + rd = \dfrac{k - rd}{1 + d}(1 + d) + rd = k$

  所以 $sG + tP_A = kG = G_1(x_1,\ y_1)$。如果$x_1'$ 和e' 没有被篡改或错误，则有$e' = e$，$x_1' = x_1$。根据产生签名算法⑤，$r = (e+x_1) \bmod n$，又根据验证签名算法⑦，$R = (e' + x_1') \bmod n$。
- 所以在$e' = e$，$x_1' = x_1$的条件下，有$R = r$ 。

武汉大学

二、中国商用公钥密码SM2签名算法

- SM2 签名验证算法的一个显著特点是，其中加入了较多的检错功能。
- 因为这是收信者对收到的签名数据进行验证，而签名数据是经过信道传输过来的，由于信道干扰和对手的篡改，因此，签名数据中含有错误或被篡改的可能性是存在的。
- 把错误和篡改检测出来，对提高签名验证系统的数据完整性、系统可靠性和安全性是有益的。
  - 验证算法中的①检查签名分量*r'*的合理性
  - 验证算法中的②检查签名分量*s'*的合理性
  - 验证算法中的⑤检查*t*的正确性

二、中国商用公钥密码SM2签名算法

6、SM2数字签名方案的应用

- 安全。
  - 目前尚没有发现求解椭圆曲线离散对数问题的亚指数算法。
- 软硬件实现规模小，方便。
  - **160位的椭圆曲线密码的安全性，相当于1024位的RSA密码。**
- 实现难点：
  - 倍点运算。
- 目前最大的应用是二代身份证。

谢　谢！