第9章 公钥密码学与RSA

1 什么是公钥密码体制

- 公钥密码又称为双钥密码和非对称密码，是1976年由Diffie和Hellman在其“密码学新方向”一文中提出的，见划时代的文献：

  W.Diffie and M.E.Hellman, New Directrions in Cryptography, IEEE Transaction on Information Theory, V.IT-22.No.6, Nov 1976, PP.644-654

单向陷门函数是满足下列条件的函数f：

(1)给定x，计算y=f(x)是容易的；

(2)给定y，计算x使y=f(x)是困难的。

(所谓计算$x=f^{-1}(Y)$困难是指计算上相当复杂;若经过数年的计算得出正确结果就无实际意义了。)

(3)存在δ，使得利用δ 对给定的函数值y，计算x使y=f(x)是容易的。

注：1*. 仅满足(1)、(2)两条的函数称为单向函数；若对第(3)条也满足,称该函数为带δ 陷门信息的单向函数,也称为陷门单向函数。

2*. 当用陷门单向函数f作为加密函数时，可将f公开，这相当于公开加密密钥。此时加密密钥便称为公开钥，记为P_k。f函数的设计者将δ 保密，用作解密密钥，此时δ 称为秘密钥匙，记为S_k。由于加密函数是公开的，任何人都可以将信息x加密成y=f(x)，然后送给函数的设计者（当然可以通过不安全信道传送）；由于设计者拥有Sk，他自然可以解出$x=f^{-1}(y)$。

3*.单向陷门函数的第(2)条性质表明窃听者由截获的密文y=f(x)推测x是计算上不可行的。

2.Diffie-Hellman密钥交换算法

Diffie和Hellman在其里程碑意义的文章中，虽然给出了密码的思想，但是没有给出真正意义上的公钥密码实例，也既没能找出一个真正带陷门的单向函数。然而，他们给出单向函数的实例，并且基于此提出Diffie-Hellman密钥交换算法。这个算法是基于有限域中计算离散对数的困难性问题之上的：设F为有限域，有$g \in F$,使F的乘法群$F^*=F\{0\}=<g>$。并且对任意正整数x，计算g^x是容易的；但是已知g和y求x使$y= g^x$，是计算上几乎不可能的。这个问题被称为有限域F上的离散对数问题。公钥密码学中使用最广泛的有限域为素域F_p.

对Diffie-Hellman密钥交换协议描述:Alice和Bob协商好一个大素数p，和大周期的整数g，$1<g<p$，g最好是F_p中的本原元，即$F_p^*=<g>$。p和g可为网络上的所有用户共享。

当Alice和Bob要进行保密通信时，他们可以按如下步骤来做：

(1)Alice选取大的随机数x，并且计算

$X=g^x(\text{mod } P)$

(2)Bob选取大的随机数x′，并计算$X' = g^{x'}(\text{mod } P)$

(3)Alice将X传送给Bob；Bob将X′传送给Alice。

(4)Alice计算$K=(X')^X(\text{mod } P)$;Bob计算$K' = (X)^{X'}(\text{mod } P)$,易见，$K=K' =g^{xx'}(\text{mod } P)$。

由(4)知，Alice和Bob已获得了相同的秘密值K。双方以K作为加解密钥以传统对称密钥算法进行保密通信。

注：**Diffie-Hellman密钥交换算法拥有美国和加拿大的专利。**

3 RSA公钥算法

RSA公钥算法是由Rivest,Shamir和Adleman在1978年提出来的（见Communitions of the ACM. Vol.21.No.2. Feb. 1978, PP.120-126)该算法的数学基础是初等数论中的Euler（欧拉)定理，并建立在分解大整数因子的困难性之上。

将Z/(n）表示为 Z_n，其中n=pq; p,q为素数且相异。若 $Z^*_n \equiv \{g \in Zn | (g,n)=1\}$，易见$Z^*_n$为 $\varphi(n)$阶的乘法群，且有 $g^{\varphi(n)} \equiv 1 (\mathrm{mod}\ n)$，而 $\varphi(n)=(p-1)(q-1)$.

RSA密码体制描述如下：

首先，明文空间P＝密文空间$C=Z_n$.加解密过程为:

(1).密钥的生成

选择p,q为互异素数，计算$n=p\times q$, $\varphi(n)=(p-1)(q-1)$, 选整数e使$(\varphi(n),e)=1, 1<e<\varphi(n))$,计算d,使$d=e^{-1}(\mathrm{mod}\ \varphi(n)))$,

公钥$P_k=\{e,n\}$;私钥$S_k=\{d,p,q\}$。

注意，当0<M<n时,有：

$M^{k\varphi(n)+1} \equiv M \pmod n$, 而$ed \equiv 1 \pmod{\varphi(n)}$,易见

$$(M^e)^d \equiv M \pmod n$$

(2).加密 (用e,n):

明文：M<n　密文：$C=M^e \pmod n$.

(3).解密 (用d,p,q)

若 密文为C ,则明文　$M=C^d$（mod n)

注：1*, 加密和解密是一对互逆运算。

2*, 对于0<M<n时，若$(M,n) \neq 1$，则M为p或q的整数倍，假设M=cp，由(cp,q)=1 有

$$M^{\varphi(q)} \equiv 1 \pmod q \Longrightarrow M^{k\varphi(q)\varphi(p)} \equiv 1 \pmod q$$

有$M^{k\varphi(q)\varphi(p)} = 1+tq$ 对其两边同乘M=cp有

有$M^{k\varphi(n)+1}=M+tcpq=M+tcn$于是

有$M^{k\varphi(n)+1} \equiv M(\mathrm{mod}\ n)$

例子：若Bob选择了p=101和q＝113，那么，n=11413, $\varphi(n)=100\times112=11200$；然而$11200=2^6\times5^2\times7$，一个正整数e能用作加密指数，当且仅当e不能被2，5，7所整除（事实上,若Bob不会分解ϕ(n)，也可试验一些数和$\varphi(n)$作辗转相除法来求得e，使（e, ϕ(n)=1)。假设Bob成功选择了e=3533，那么继续用辗转相除法可以求得他的解密钥 $d=e^{-1} \equiv 6597(\mathrm{mod}\ 11200)$.

Bob在一个目录中公开n=11413和e=3533, 现假设Alice想发送明文9726给Bob，她计算密文：

$$9726^{3533}(\mathrm{mod}\ 11413)=5761$$

并且在一个信道上发送密文5761。当Bob接收到密文5761时，他用他的秘密解密指数（私钥）d＝6597进行解密：5761^{6597}（mod 11413)=9726

注：RSA的安全性是基于加密函数$e_k(x)=x^e(\text{mod } n)$是一个单向函数，所以对那些没有陷门信息的人来说,求逆计算不可行。而Bob掌握φ (n)=(p-1)(q-1),从而用欧氏算法可以解出解密私钥d.

4 RSA密码体制的实现

实现的步骤如下：Bob为实现者

(1)Bob寻找出两个大素数p和q

(2)Bob计算出n=pq和φ (n)=(p-1)(q-1).

(3)Bob选择一个随机数e(0<e< φ (n))，满足（e， φ (n)）＝1

(4)Bob使用辗转相除法计算$d=e^{-1}(\mod \varphi(n))$

(5)Bob在目录中公开n和e作为她的公开钥。

密码分析者攻击RSA体制的关键点在于如何分解n。若分解成功使n=pq，则可以算出ϕ(n)＝（p-1)(q-1)，然后由公开的e，解出秘密的d。（猜想：攻破**RSA**与分解**n**是多项式等价的。然而，这个猜想至今没有给出可信的证明！！！）

于是要求：若使RSA安全，p与q必为足够大的素数，使分析者没有办法在多项式时间内将n分解出来。建议选择p和q大约是100位的十进制素数。 模n的长度要求至少是512比特。EDI攻击标准使用的RSA算法中规定n的长度为512至1024比特位之间，但必须是128的倍数。国际数字签名标准ISO/IEC 9796中规定n的长度位512比特位。

为了抵抗现有的整数分解算法，对RSA模n的素因子p和q还有如下要求：

(1)|p-q|很大，通常 p和q的长度相同；

(2)p-1 和q-1分别含有大素因子p_1和q_1

(3)p_1-1和q_1-1分别含有大素因子p_2和q_2

(4)p+1和q+1分别含有大素因子p_3和q_3

为了提高 加密速 度，通常取e为特定的小整数，如EDI国际标准中规定 $e=2^{16}+1$，ISO/IEC9796中甚至允许取 $e=3$。这时加密速 度一般比解密速 度快10倍以上。

下面研究加解密算术运算，这个运算主要是模n的求幂运算。著名的“平方-和-乘法”方法将计算 $x^c \pmod n$ 的模乘法的数目缩小到至多为 $2l$，这里的 l 是指数c的二进制表示比特数。若设n以二进制形式表示有k比特，即 $k=[\log_2 n]+1$。 由 $l \le k$，这样 $x_c \pmod n$ 能在 $o(k^3)$ 时间内完成。（注意，不难看到，乘法能在 $o(k^2)$ 时间内完成。）

平方－和－乘法算法：

指数c以二进制形式表示为：

$$c=\sum_{i=0}^{t-1} c_i 2_i = c_0+c_1 2+\ldots+c_{t-1}2_{t-1} \quad c_i \in \{0,1\}$$

$X^c=x^{c0}\times(x^2)^{c1}\times\ldots\times(x^{2^{t-1}})^{c_{t-1}}$

预计算：$x^2=xx$

$x^4=x^{2^2}=x^2x^2$

.

.

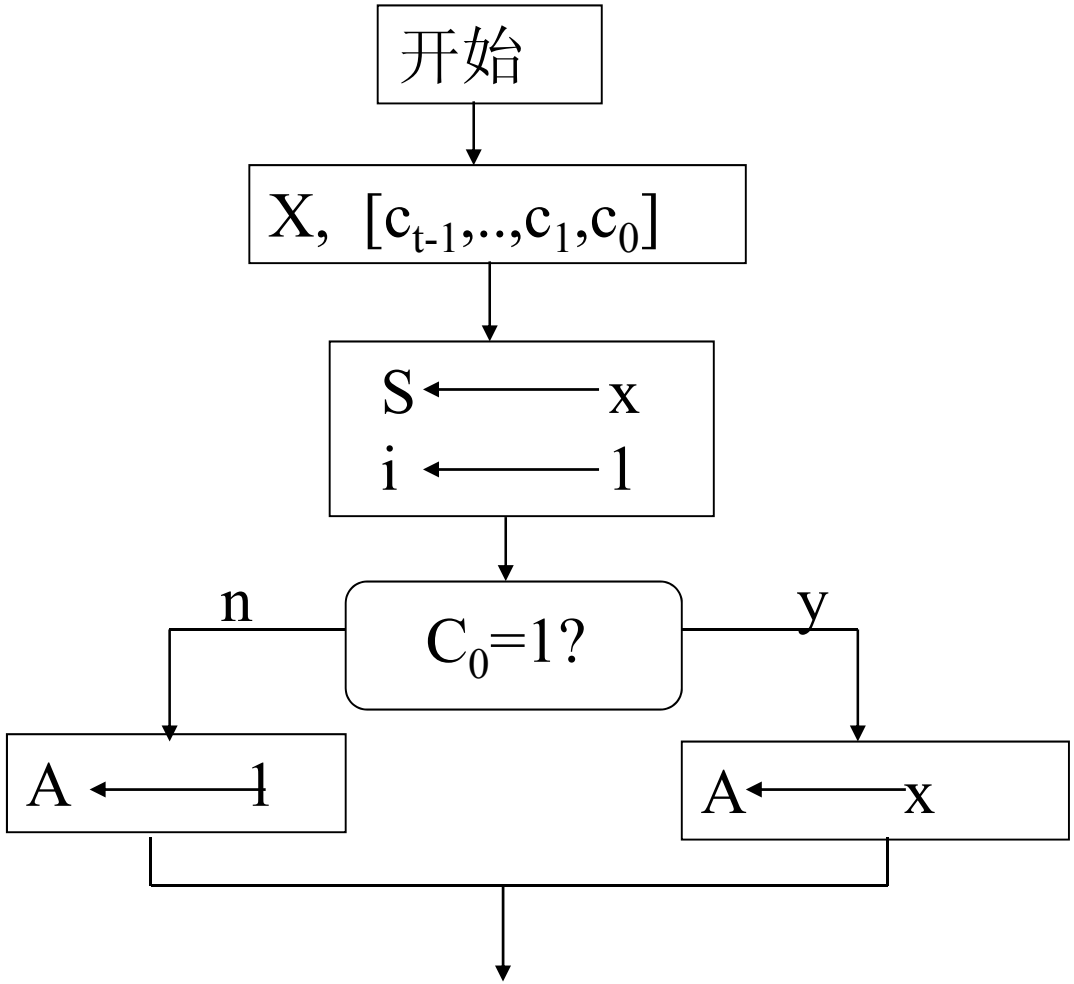
.

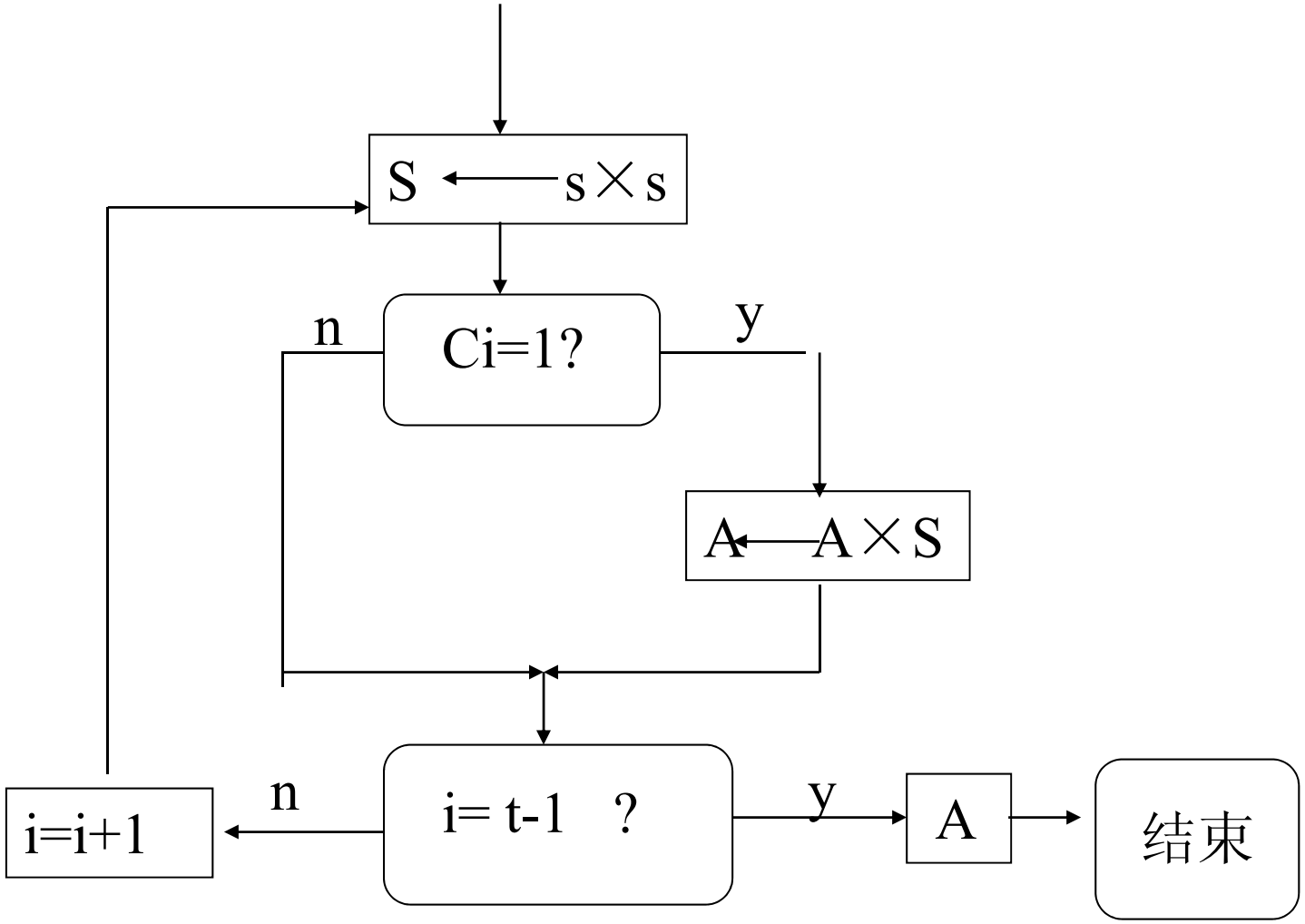
$x^{2^{t-1}}=x^{2^{t-2}}*x^{2^{t-2}}$

} t-1次乘法

X^c计算：把那些$c_i=1$对应的x^{2^i}全部乘在一起，便得x^c。至多用了t-1次乘法。请参考书上的177页，给出计算$x^c(\text{mod } n)$算法程序：

$A=x^c \quad c=c_0+c_1 2+..+c_{t-1}2^{t-1}=[c_{t-1},....,c_1,c_0]_2$


开始
X, [c_{t-1},..,c_1,c_0]
S ← x
i ← 1
n
C_0=1?
y
A ← 1
A ← x


S ← s×s
Ci=1?
n
y
A ← A×S
i= t-1 ?
n
y
i=i+1
A
结束

5 RSA签名方案

签名的基本概念

传统签名（手写签名）的特征：

（1）一个签名是被签文件的物理部分；

（2）验证物理部分进行比较而达到确认的目的。(易伪造)

（3）不容易忠实地“copy”!!!

定义：(数字签名方案）一个签名方案是有签署算法与验证算法两部分构成。可由五元关系组（P,A,K,S,V）来刻化：

(1)P是由一切可能消息(messages)所构成的有限集合；

(2)A是一切可能的签名的有限集合；

(3)K为有限密钥空间，是一些可能密钥的有限集合；

(4)任意k ∈K，有签署算法Sig_k ∈ S且有对应的验证算法 Ver_k∈V,对每一个

Sig_k:p → A 和Ver_k:P×A {真，假}

满足条件：任意x∈ P,y∈ A.有签名方案的一个签名：

$$Ver(x,y) = \begin{cases} 真 & 若y=sig(x) \\ 假 & 若y\neq Sig(x) \end{cases}$$

注：1*.任意k∈K, 函数Sig_k和Ver_k都为多项式时间函数。

2*.Ver_k为公开的函数，而Sig_k为秘密函数。

3*.如果坏人（如Oscar)要伪造Bob的对X的签名，在计算上是不可能的。也即,给定x，仅有Bob能计算出签名y使得$Ver_k(x,y)$＝真。

4*.一个签名方案不能是无条件安全的，有足够的时间使Oscar总能伪造Bob的签名。

RSA签名：n=pq,P=A=Zn,定义密钥集合

$K=\{(n,e,p,q,d)\}|n=pq,de\equiv 1(\mod \varphi(n))\}$

注意：n和e为公钥；p,q,d为保密的（私钥）。对$x \in P$,
Bob要对x签名，取$k \in K$。$Sig_k(x) \equiv x^d (\mathrm{mod}\ n) \equiv y (\mathrm{mod}\ n)$

于是

$Ver_k(x,y)$=真 $\Longleftrightarrow$ $x \equiv y^e (\mathrm{mod}\ n)$

（注意：**e,n公开；可公开验证签名(x,y)对错！！也即是否为Bob的签署**）

注：1*.任何一个人都可对Bob的一个签署y,利用Bob的公钥来计算$x=?e_k(y)$，来验证Bob对消息x是否真的签名。

2*.签名消息的加密传递问题：假设Alice想把签了名的消息加密送给Bob，她按下述方式进行：对明文x，Alice计算对x的签名，$y=Sig_{Alice}(x)$，然后用Bob的公开加密函数e_{Bob},计算出 $Z=e_{Bob}(x,y)$,Alice 将Z传给Bob，Bob收到Z后，第一步解密：

$d_{Bob}(Z)=d_{Bob}e_{Bob}(x,y)=(x,y)$

然后用Alice公钥检验

$Ver_{Alice}(x,y)=$ 真

问题：若Alice首先对消息x进行加密，然后再签名，结果如何呢？$Y=Sig_{Alice}(e_{Bob}(x))$,注意 $z=e_{Bob}(x)$.

Alice 将（z,y)传给Bob，Bob先将z解密，获取x;然后用Ver_{Alice}检验关于x的加密签名y。这个方法的一个潜在问题是，如果Oscar获得了这对（z,y)，他能用自己的签名来替代Alice的签名

$y'=Sig_{Oscar}(e_{Bob}(x))$, $z=z'=e_{Bob}(x)$

(注意：Oscar能签名密文$e_{Bob}(x)$，甚至他不知明文x也能做。Oscar传送（z,y ′)给Bob,Bob可能推断明文x来自Oscar。所以，至今人们还是推荐先签名后加密。）

6.EIGamal方案

EIGamal公钥密码体制是基于求解离散对数困难问题的.
设P至少是150位的十进制素数，p-1有大素因子。Z_p为有域，
若α为Z_p中的本原元，有Z_p^*＝<α>。若取β∈Z_p^*=Z_p\{0}，
如何算得一个唯一得整数a,（要求，0≤a≤ p-2)，满足

$\alpha^a=\beta(\mathrm{mod}\ p)$

将a记为$a=\log_\alpha\beta$

一般来说，求解a在计算上是难处理的。

Z_p^*中的EIgamal公钥体制的描述：设明文空间为P=Z_p^*,密文空间为C=$Z_p^*\times Z_p^*$,定义密钥空间K={(p, α,a, β)|$\beta=\alpha^a(\mathrm{mod}\ p)$}
公开钥为：p, α ,β
秘密钥（私钥）：a

Alice 取一个秘密随机数$k \in Z_{p-1}$，对明文x加密

$$e_k(x,k)=(y_1,y_2)$$

其中， $y_1=\alpha^k \pmod p$, $y_2=x\beta^k \pmod p$

Bob解密，

$$d_k(y_1,y_2)=y_2(y_1^{\alpha})^{-1} \pmod p$$

注：1*.容易验证$y_2(y_1^{\alpha})^{-1}=x(\alpha^a)^k(\alpha^{ka})^{-1}=x$!!

2*.利用EIGamal加密算法可给出基于此的签名方案：

Bob要对明文x进行签名，他首先取一个秘密随机数k作为签名

$$Sig_k(x,k)=(\gamma , \delta)$$

其中 $\gamma=\alpha^k \pmod p$, $\delta=(x-a\gamma)k^{-1} \pmod{p-1}$

对$x, \gamma \in Z_p^*$和$\delta \in Z_{p-1}$，定义$Ver_k(x, \gamma,\delta)$＝真，它等价于

$$\beta^{\gamma}\gamma^{\delta}=\alpha^{x} \pmod p$$

要说明的是，如果正确地构造了这个签名，那么验证将是成功的，因为

$$\beta^{\gamma}\gamma^{\delta} = \alpha^{a\gamma}\alpha^{k\delta} \pmod p = \alpha^{a\gamma+k\delta} \pmod p$$

由上面知道，$\delta = (x - a\gamma)k^{-1} \pmod{p-1}$可以推出

$k\delta = x - a\gamma \pmod{p-1}$有$a\gamma + k\delta \equiv x \pmod{p-1}$

所以 $\beta^{\gamma}\gamma^{\delta} = \alpha^{x} \pmod p$

该签名方案已经被美国NIST(国家标准技术研究所）确定为签名标准（1985）。

有关RSA方面的内容，请访问网址：

www.RSAsecurity.com