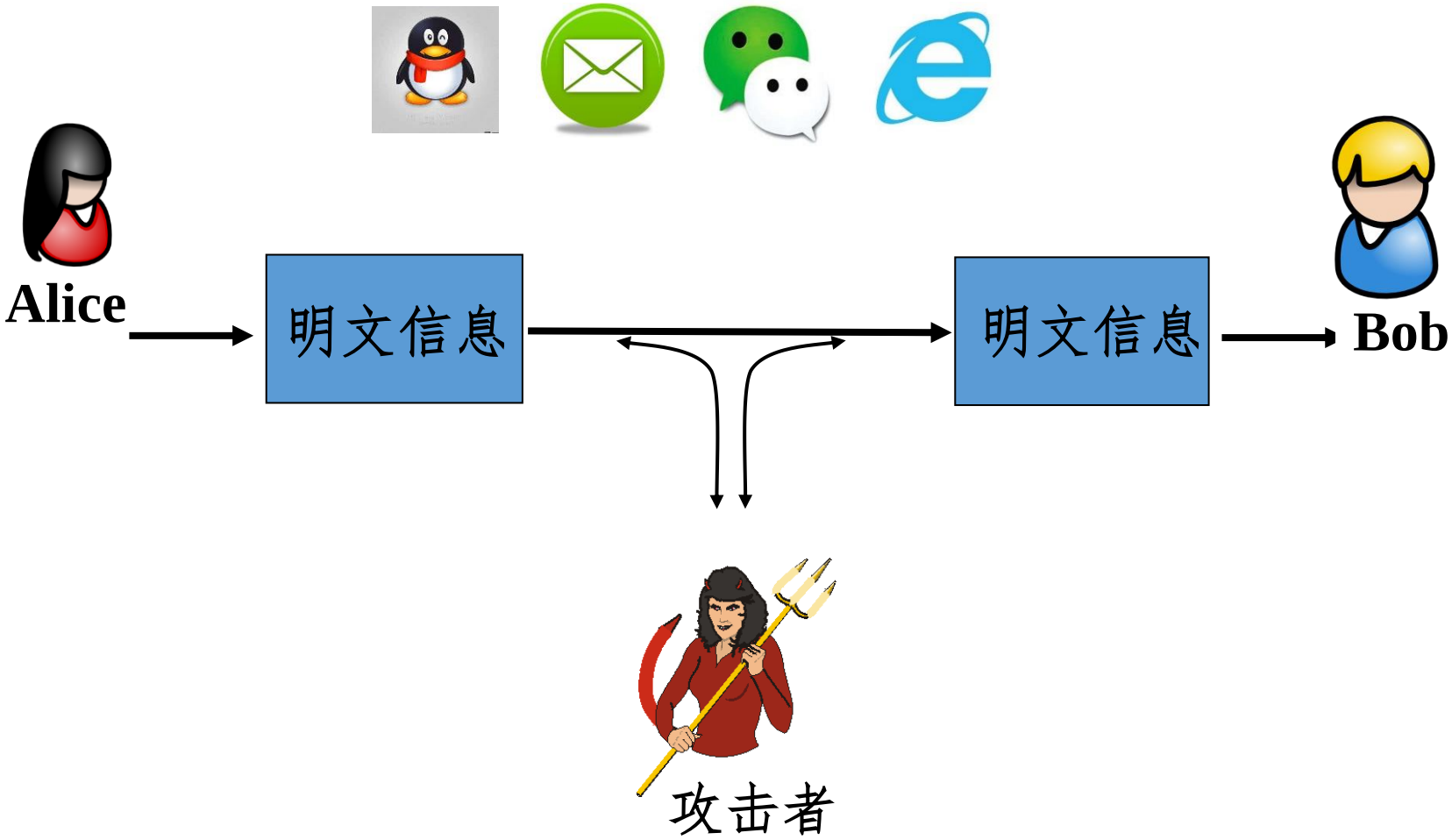
现代密码学

第四章 公钥密码

信息与软件工程学院

公开信道上的“安全”通信



“传统的”对称密码体制无法解决上述问题
--------密钥协商

公钥密码体制的概念是在解决单钥密码体制中最难解决的两个问题时提出的，这两个问题是密钥分配和数字签名。

公钥密码的概念是谁提出的？

- A. **Rivest, Shamir, Adleman** — 第一个公钥加密算法的提出者
- B. **Diffie, Hellman** — 公钥密码概念的提出者
- C. **Joan Daemen, Vincent Rijmen** — **AES的设计者**
- D. **Shannon** — 密码学由艺术转变为科学的推动者

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D. Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enci-

公钥密码体制

- 主要思想：
  - 一些问题呈现出“非对称性”- 从一个方向计算非常容易，而从另一个方向计算则很困难
  - 例如：计算任意给定整数的乘积很容易，而计算给定大整数的因子则非常困难

公钥密码的理论基础：　陷门单向函数

单向函数的例子？

单向函数：　已知x，计算y使得y=f(x)容易；

已知y，计算x使得y=f(x)是难解的。

陷门单向函数：　t是与f有关的一个参数；已知x，计算y使得y=f(x)容易；

(1)不知道t，已知y，计算x使得y=f(x)是难的，

(2)知道t时，已知y，计算x使得y=f(x)是容易的。

参数t称为陷门。

公钥密码体制

- 每个用户生成一个密钥对：一个公钥**pk**和一个对应的私钥 **sk**
  - 公钥将在系统内被公开**(PKI) public key infrastructure**
  - 私钥由用户本人安全保管

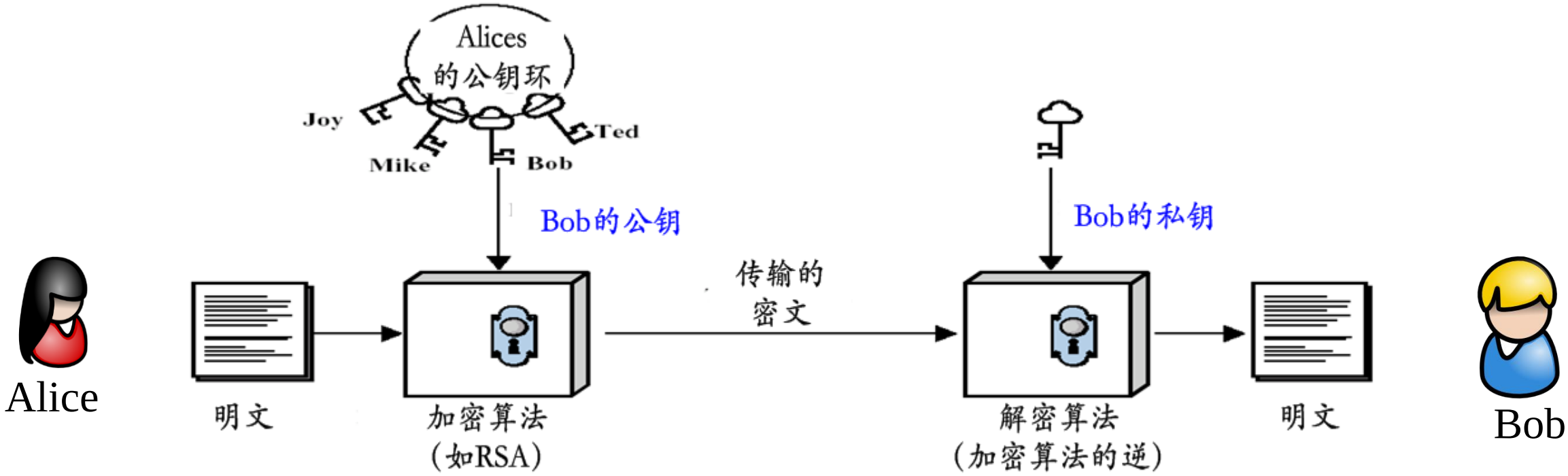
- 私钥由用户本人使用，而公钥则由系统中其他用户使用

- 公钥密码体制也被称为： 非对称密码体制

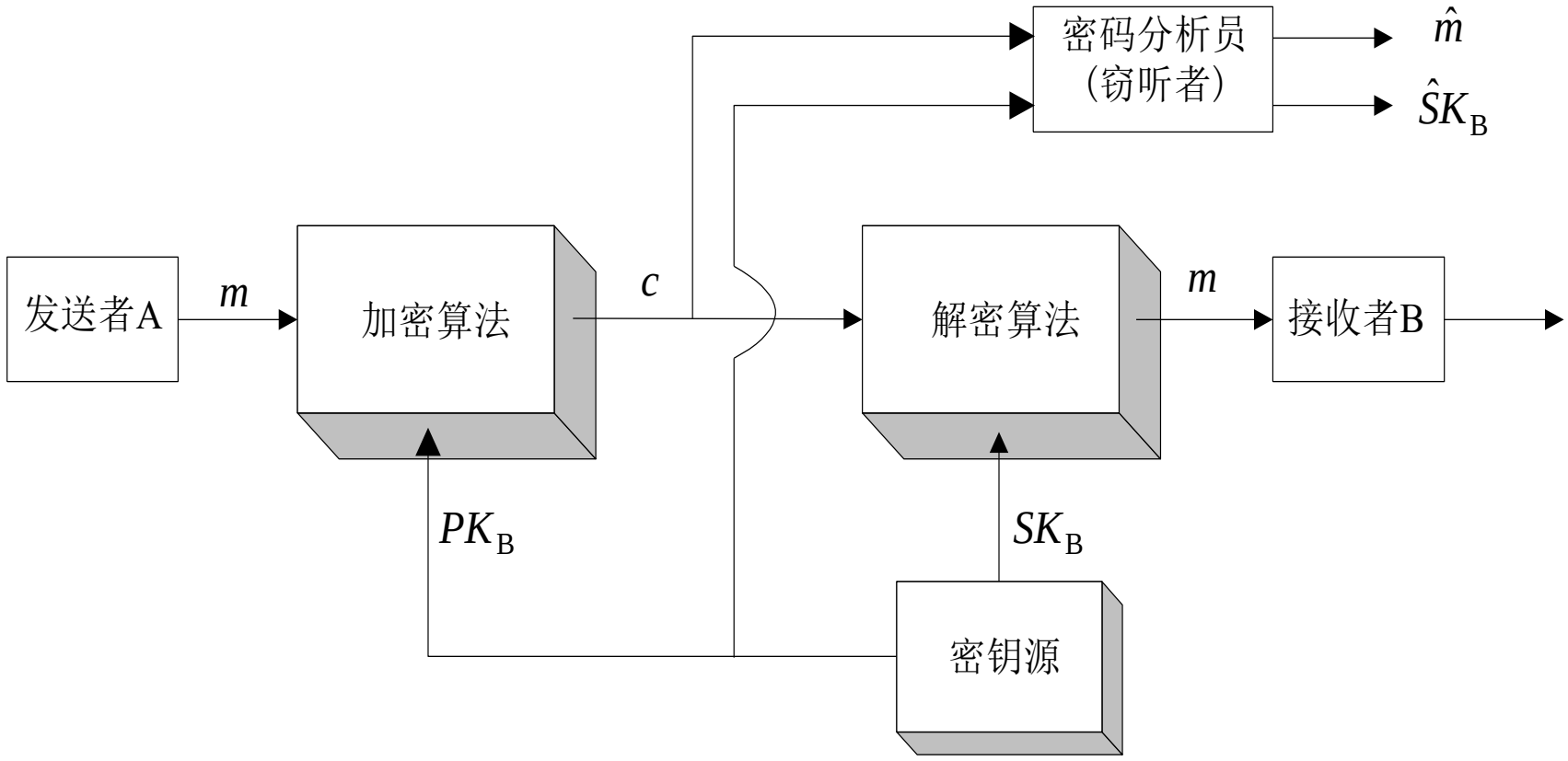
公钥密码体制之基于身份的密码体制

- 每个用户生成一个密钥对：一个身份（**ID**，公钥）和一个对应的私钥 **sk**
  - 身份就是公钥
  - 私钥由用户本人安全保管
- 私钥**sk**需要可信第**3**方产生---密钥生成中心（**KGC**）

公钥密码体制的基本思想

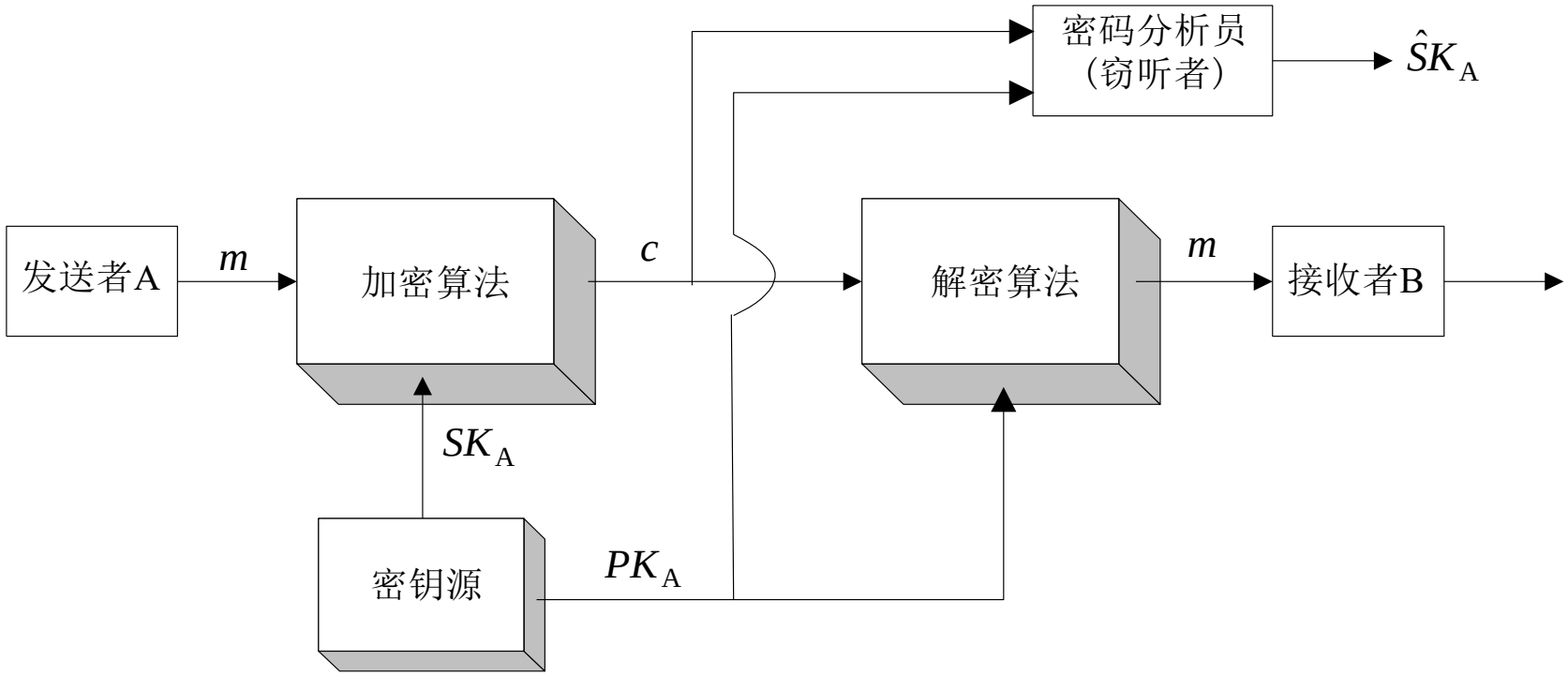


公钥加密



公钥体制加密的框图

公钥认证



公钥密码体制认证框图

公钥密码体制的基本概念

- 对称密码算法的缺陷
  - 密钥分配问题：通信双方加密通信前要通过秘密的安全信道协商加密密钥，这种安全信道可能很难实现；对这个信道安全性的要求比正常传送消息信道的安全性要高
  - 密钥管理问题：在多用户网络中，任何两个用户之间都需要有共享的秘密钥，n个用户需要$C_n^2=n(n-1)/2$个密钥，n=5000时，$C_n^2=12,497,500$，系统开销非常大
  - 没有签名功能：当主体A收到主体B的电子文挡时，无法向第三方证明此电子文档确实来源于B，传统单钥加密算法无法实现抗抵赖的需求

公钥密码算法应满足的要求

•公钥密码算法应满足以下要求：

•① 公私钥生成容易：接收方B产生密钥对（PK_B和SK_B）在计算上是容易的。

•② 加密容易：发方A加密m以产生密文c，即$c=E_{PKB}[m]$在计算上是容易的。

•③ 解密容易：收方B用自己的秘密钥对c解密，即$m=D_{SKB}[c]$计算上是容易的。

•④ 计算私钥难：敌手由PK_B求SK_B在计算上是不可行的。

•⑤ “解密”难：敌手由密文c和PK_B恢复明文m在计算上是不可行的。

•以上要求的本质之处在于要求一个陷门单向函数。

•研究公钥密码算法就是要找出合适的陷门单向函数。

计算复杂性

对一个密码系统来说，应要求在密钥已知的情况下，加密算法和解密算法是“容易的”，而在未知密钥的情况下，推导出密钥和明文是“困难的”。那么如何描述一个计算问题是“容易的”还是“困难的”？

可用解决这个问题的算法的计算时间和存储空间来描述。

算法的计算时间和存储空间（分别称为算法的时间复杂度和空间复杂度）定义为算法输入数据的长度 n 的函数 $f(n)$ 。

当 n 很大时，通常只关心 $f(n)$ 随着 n 的无限增大是如何变化的，即算法的渐近效率。

O 记号

O 记号给出的是 $f(n)$ 的渐近上界。如果存在常数 C 和 N ，当 $n>N$ 时，有 $f(n)\le Cg(n)$ ，则记 $f(n)=O(g(n))$ 。所以 O 记号给出的是 $f(n)$ 在一个常数因子内的上界。

例如，$f(n)=8n+10$ ，则当 $n>N=10$ 时，$f(n)\le 9n$ ，所以 $f(n)=O(n)$ 。

一般，若 $f(n)=a_0+a_1n+\cdots+a_kn^k$，则 $f(n)=O(n^k)$ 。

若算法的时间复杂度为 $T=O(n^k)$ ，则称该算法是多项式时间的；
若 $T=O(k^{f(n)})$ ，其中 k 是常数，$f(n)$ 是多项式，就称该算法是指数时间的。

P和NP问题

字母表Σ是一个有限的符号集合，Σ上的语言L是 Σ 上的符号构成的符号串的集合。

一个图灵机 M 接受一个语言L表示为 $x \in L \Leftrightarrow M(x)=1,$
这里简单地用 1来表示接受。

有两种类型的计算性问题是比较重要的。
第一种是可以在多项式时间内判定的语言集合，表示为 P 。
正式地说， $x \in L$ ，当且仅当存在图灵机在最多$p(|x|)$（p为某个多项式，x 是图灵机的输入串，$|x|$ 表示x 的长度）步内接受一个输入x ，我们就说语言 L 在 P 中。

P和NP问题

第二种是**NP**类语言，**NP**问题是指可在多项式时间内验证它的一个解的问题。即对语言中的元素存在多项式时间的图灵机可验证该元素是否属于该语言。

正式地说，如果存在一个多项式图灵机 M 使得 $x \in L$ 当且仅当存在一个串 w_x 使得 $M(x, w_x) = 1$。我们就说语言 $x \in L$ 在**NP**中。w_x 称为 ***x*** 的证据，用于证明 L。

因为可在多项式时间内求解就一定可在多项式时间内验证。但反过来不成立，因为求解比验证解更为困难。

用 ***P*** 表示所有 ***P*** 问题的集合，**NP**表示所有**NP**问题的集合，则有 $\mathrm{P} \subset \mathrm{NP}$。

在**NP**类中，有一部分可以证明比其他问题困难，这一部分问题称为**NPC**问题。也就是说，**NPC**问题是**NP**类中“最难”的问题。

几个困难问题

- （1）大整数分解问题（factorization problem）

若已知两个大素数p和q，求$n = pq$是容易的，只需一次乘法运算，而由n，求p和q则是困难的，这就是大整数分解问题。

- （2）离散对数问题（discrete logarithm problem）

给定一个大素数p，$p-1$含另一大素数因子q，则可构造一个乘法群，它是一个$p-1$阶循环群。设g是的一个生成元，$1<g<p-1$。已知x，求$y=g^x \bmod p$是容易的，而已知y、g、p，求x使得$y=g^x \bmod p$成立则是困难的，这就是离散对数问题。

- （3）多项式求根问题

有限域$GF(p)$上的一个多项式：

$$y = f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \bmod p$$

已知 $a_0, a_1, \ldots, a_{n-1}$，$p$和$x$，求$y$是容易的，而已知$y, a_0, a_1, \ldots, a_{n-1}$，求$x$则是困难的，这就是多项式求根问题。

- （4）二次剩余问题（quadratic residue problem）

给定一个合数n和整数a，判断a是否为mod n的二次剩余，这就是二次剩余问题。在n的分解未知时，求 $x^2 \equiv a \bmod n$的解也是一个困难问题。

- （5）背包问题（**knapsack problem**）

给定向量$A=(a_1,a_2,\ldots,a_n)$（a_i为正整数)和$x=(x_1,x_2,\ldots,x_n)$（$x_i\in\{0,1\}$），求和式：

$$s=f(x)=a_1x_1+a_2x_2+\cdots+a_nx_n$$

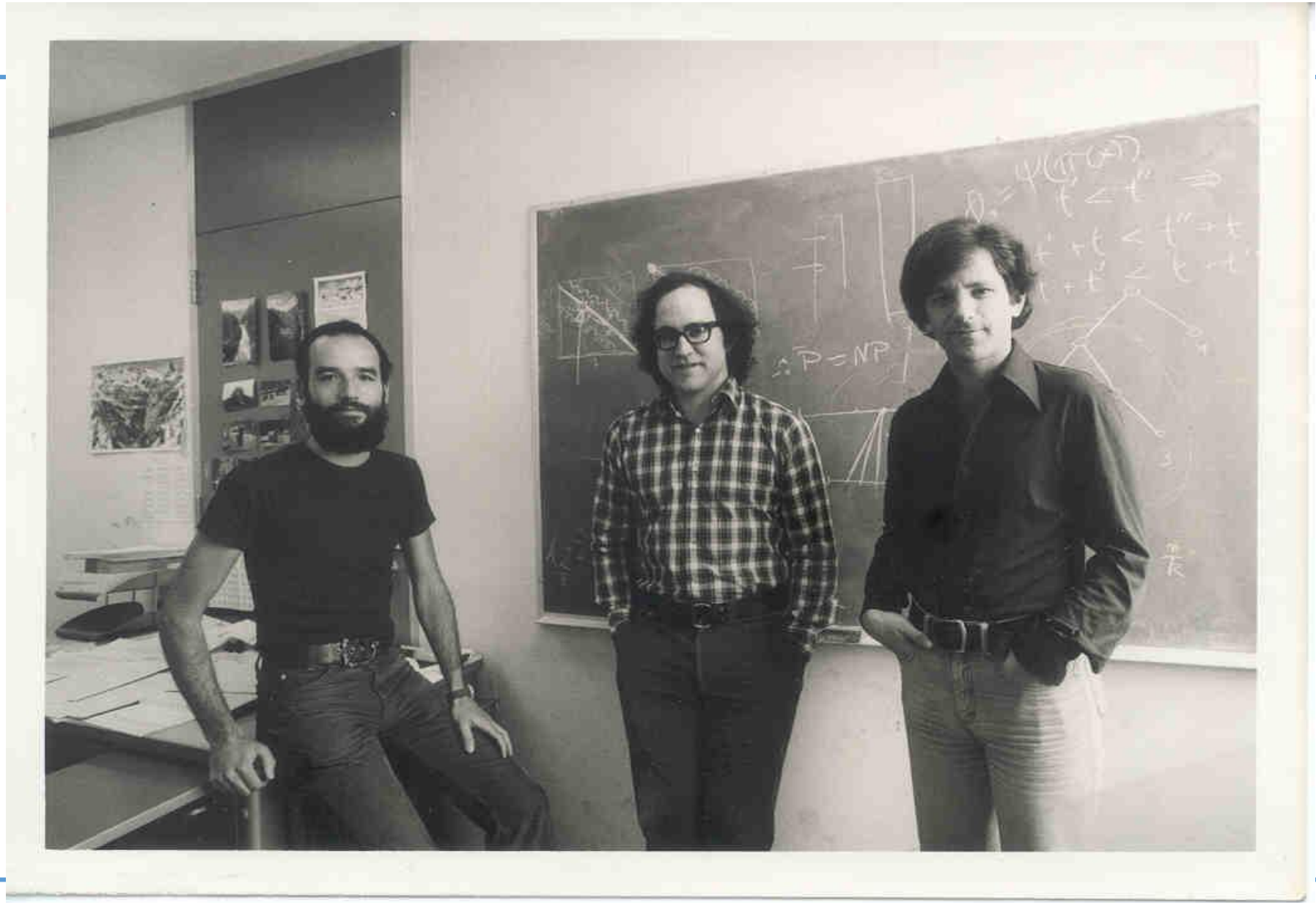
是容易的，而由A和S，求x则是困难的，这就是背包问题，又称子集和问题。

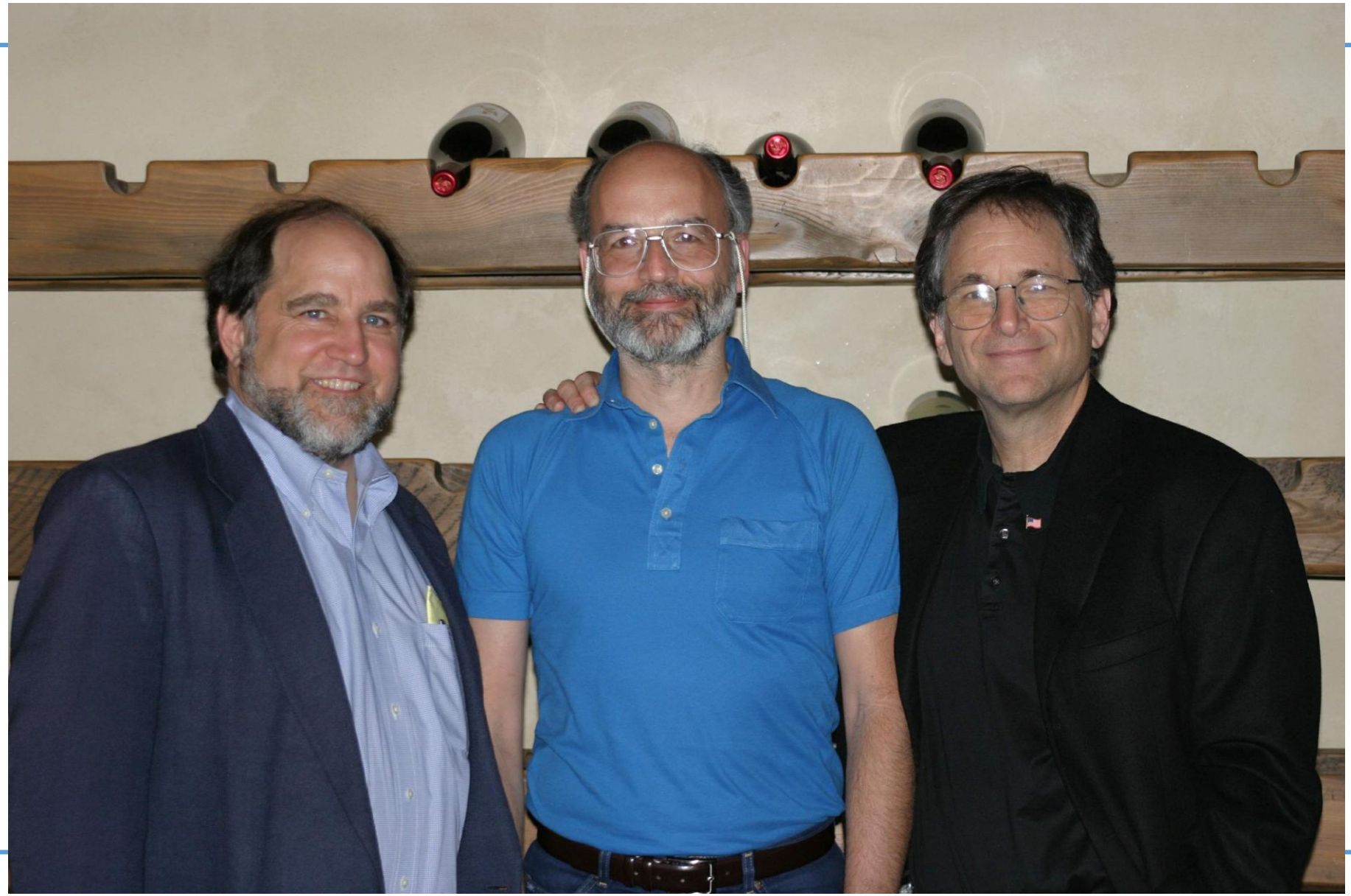
- （6）MQ 问题
  - 求解有限域上任意选取的多变量二次多项式方程组问题

$$\begin{cases} y_1' = p_1(x_1, \ldots, x_n) \\ \vdots \\ y_m' = p_m(x_1, \ldots, x_n) \end{cases}$$

RSA公钥密码算法

- MIT三位年青学者Rivest，Shamir和Adleman 在1978年发现了一种用数论构造双钥体制的方法，称作MIT体制，后来被广泛称之为RSA体制。
- 它既可用于加密、又可用于数字签名。
- RSA算法的安全性基于数论中大整数分解的困难性。


P=NP



RSA算法描述------密钥生成

- 选取两互异大素数p和q
- 计算 $n=p\times q$和其欧拉函数值$\varphi(n)=(p-1)(q-1)$
- 选一整数e，$1<e<\varphi(n)$，使得$\gcd(\varphi(n), e)=1$
- 在模$\varphi(n)$下，计算e的逆元d. 即求d, 使得

$$ed \equiv 1 \bmod \varphi(n)$$

- 以(n, e)为公钥。d为秘密钥。

  (p, q不再需要，可以销毁。)

RSA算法描述------密钥生成（简化版）

- $n = pq$，大素数p和q
- 选公钥e，（其中$1 < e < \varphi(n)$，使得 $\gcd(\varphi(n), e)=1$）
- 计算私钥d（$ed \equiv 1 \bmod \varphi(n)$）

- 加密

  将明文分组，各组对应的十进制数$m<n$，计算

$$c = \mathrm{E}(m) \equiv m^{e} \bmod n$$

- 解密　　$m = \mathrm{D}(c) \equiv c^{d} \bmod n$

•RSA算法解密过程的正确性。

•证明： 由加密过程知$c \equiv m^e \bmod n$，所以

•$c^d \bmod n \equiv m^{ed} \bmod n \equiv m^{k\phi(n)+1} \bmod n$

加密： $c \equiv m^e \bmod n$

解密： $m \equiv c^d \bmod n$

分两种情况：

①**m与n互素**，则由**Euler**定理得

$$m^{\varphi(n)} \equiv 1 \bmod n,$$

$$m^{k\varphi(n)} \equiv 1 \bmod n,$$

$$m^{k\varphi(n)+1} \equiv m \bmod n$$

即$c^d \bmod n \equiv m$。

② gcd(m,n)≠1,

不妨设 m=cp， $1 \le m < n, 1 \le c < q$

由gcd(m,q)=1及Euler定理得$m^{\varphi(q)} \equiv 1 \bmod q$，所以

$$m^{k\varphi(q)} \equiv 1 \bmod q,$$

$$[m^{k\varphi(q)}]^{\varphi(p)} \equiv 1 \bmod q,$$

$$m^{k\varphi(n)} \equiv 1 \bmod q$$

因此存在一整数r，使得$m^{k\varphi(n)}=1+rq$，两边同乘以m=cp得

$$m^{k\varphi(n)+1}=m+rcpq=m+rcn$$

即$m^{k\varphi(n)+1} \equiv m \bmod n$，所以$c^d \bmod n \equiv m$。(证毕)

问题：

- 虽然我们上面讨论了m与n互素和m与n不互素2种情况，
- 但在实际应用过程中， m与n是互素，为什么？？？

例

- 例： p=7， q=17. n=p×q=119， Φ(n)=(p-1)(q-1)=96。取e=5，满足1<e<Φ(n)，且gcd(Φ(n),e)=1。
- 确定满足d·e=1 mod 96且小于96的d，因为77×5=385=4×96+1，故d=77。
- 因此公钥为{5，119}，私钥为{77，119}。
- 设明文m=19，则由加密过程得密文为
- $c \equiv 19^5 \mod 119 \equiv 2476099 \mod 119 \equiv 66$
- 解密为
- $66^{77} \mod 119 \equiv 19$

RSA算法中的计算问题

•1. RSA的加密与解密过程

• RSA的加密、解密过程都为求一个整数的整数次幂，再取模。如果按其含义直接计算，则中间结果非常大，有可能超出计算机所允许的整数取值范围。而用模运算的性质：

$$(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$$

•就可减小中间结果。

平方和乘法

- 再者，考虑如何提高加、解密运算中指数运算的有效性。例如求x^{16}，直接计算的话需做15次乘法。然而如果重复对每个部分结果做平方运算即求x，x^2，x^4，x^8，x^{16}则只需4次乘法。
- 求a^m可如下进行，其中a，m是正整数：
- 将m表示为二进制形式$b_k\ b_{k-1}...b_0$，即
- 因此

$$a^m = (\cdots(((a^{b_k})^2 a^{b_{k-1}})^2 a^{b_{k-2}})^2 \cdots a^{b_1})^2 a^{b_0}$$

$$m=b_k2^k+b_{k-1}2^{k-1}+\ldots+b_12+b_0$$

$$a^m = (\cdots(((a^{b_k})^2 a^{b_{k-1}})^2 a^{b_{k-2}})^2 \cdots a^{b_1})^2 a^{b_0}$$

快速指数算法

```
d=1;
For  i=k downto 0 do {
        d=(d×d) mod n;
        if b_i=1 then    {
                d=(d×a) mod n
                        }
        }
return d.
```

- 2. RSA密钥的产生

(1) 两个大素数p、q的选取。

- Miller-Rabin算法;

- (2) e的选取和d的计算。
- 选取满足$1<e<\Phi(n)$和$\gcd(\Phi(n),e)=1$的e，并计算满足$d\cdot e\equiv 1 \bmod \Phi(n)$的d。这一问题可由推广的Euclid算法完成。

素性检验

•素性检验是指对给定的数检验其是否为素数。对于大数的素性检验来说没有简单直接的方法，本节介绍一个概率检验法，为此需要以下引理。

•引理 如果p为大于2的素数，则方程$x^2 \equiv 1 \pmod p$的解只有$x \equiv 1$和$x \equiv -1$。

证明：由$x^2 \equiv 1 \bmod p$，

有$x^2-1 \equiv 0 \bmod p$，$(x+1)(x-1) \equiv 0 \bmod p$，

因此$p|(x+1)$或$p|(x-1)$。

设$p|(x+1)$，则$x+1=kp$，

因此$x \equiv -1 \pmod p$。

类似地可得 $x \equiv 1 \pmod p$。（证毕）

•引理的逆否命题为：如果方程$x^2 \equiv 1 \mod p$有一解$x_0 \notin \{-1,1\}$，那么p不为素数。

例如： 方程$x^2 \equiv 1 \pmod 8$的解为1，-1，3，-3，可见8不是素数。

Miller-Rabin的素性概率检测法基于的结论：

•定理4.2 (Fermat)若p是素数，a是正整数且gcd(a，p)=1，则$a^{p-1}\equiv 1 \bmod p$。

引理 如果p为大于2的素数，则方程$x^2\equiv 1 \pmod p$的解只有$x\equiv 1$和$x\equiv -1$。

Miller-Rabin的素性概率检测法（判断n是否为素数）

•Miller-Rabin的素性概率检测法（判断n是否为素数）。首先将n-1表示为二进制形式$b_k b_{k-1}...b_0$，并给d赋初值1，则算法Witness(a,n)的核心部分如下：（令m=n-1， a是小于n的整数）

将m=n-1表示为二进制形式$b_k\ b_{k-1}...b_0$，

•for i=k downto 0 do

•　　{

•　　x←d;

•　　d←(x×x) mod n;

•　　if d=1 and(x≠1)and(x≠n-1)then return *False*;

•　　if b_i=1 then d←(d×a) mod n

•　　}

•if d≠1 then return False;

•return *True*.

$$a^m = (\cdots(((a^{b_k})^2 a^{b_{k-1}})^2 a^{b_{k-2}})^2 \cdots a^{b_1})^2 a^{b_0}$$

Miller-Rabin的素性概率检测法

100	0.789×10^{-30}

图 4.8 Solovay—Strassen 测试

是素数，我们将证明这个算法不能回答“n 是合数

1. 写 $n-1=2^k m$，m 是奇数
2. 选择一个随机整数 a，$1\leqslant a\leqslant n-1$
3. 计算 $b=a^m \bmod n$
4. if $b\equiv 1 \pmod n$ then
   回答“n 是素数”，退出
5. for $i=0$ to $k-1$ do
6. if $b\equiv -1 \pmod n$ then
   回答“n 是素数”，退出
   else $b=b^2 \bmod n$
7. 回答“n 是合数”

图 4.9 对奇数 n 的 Miller—Rab

4.10：合数的 Miller—Rabin 算法是一个 yes—bias

算法解释

•此算法输入：n，a（<n）。算法输出：(1)False，则n肯定不是素数；

•　　(2) True，则n有可能是素数。

算法解释

•此算法输入：n，a（<n）。算法输出：(1)False，则n肯定不是素数；

•　　(2) True，则n有可能是素数。

•for循环结束后，有$d \equiv a^{n-1} \bmod n$，由Fermat定理知，若n为素数，则d为1。因此若$d \neq 1$，则n不为素数，所以返回*False*。

算法解释

- 此算法输入：n，a（<n）。算法输出：(1)False，则n肯定不是素数；
- (2) True，则n有可能是素数。
- for循环结束后，有$d \equiv a^{n-1} \bmod n$，由Fermat定理知，若n为素数，则d为1。因此若$d \neq 1$，则n不为素数，所以返回*False*。
- 该算法有以下性质： 对s个不同的a，重复调用这一算法，只要有一次算法返回为*False*，就可肯定n不是素数。如果算法每次返回都为*True*，则n是素数的概率至少为$1-2^{-s}$，因此对于足够大的s，就可以非常肯定地相信n为素数。
- 为什么会出错？

算法解释

•此算法输入：n，a（<n）。算法输出：(1)False，则n肯定不是素数；

•　　(2) True，则n有可能是素数。

•for循环结束后，有$d \equiv a^{n-1} \bmod n$，由Fermat定理知，若n为素数，则d为1。因此若$d \neq 1$，则n不为素数，所以返回*False*。

•该算法有以下性质：　对s个不同的a，重复调用这一算法，只要有一次算法返回为*False*，就可肯定n不是素数。如果算法每次返回都为*True*，则n是素数的概率至少为$1-2^{-s}$，因此对于足够大的s，就可以非常肯定地相信n为素数。

•为什么会出错？

•n本身是合数，但是a使得$1=d \equiv a^{n-1} \bmod n$成立。即**ord(a)| n-1,**
推出 **gcd(\phi, n-1) =2k**

RSA解密的CRT方法

- n=pq，解密指数d与n同等规模
- 要计算$m = c^d$ （mod n）　　（*）
- 计算$d_p = d$（mod p-1），d_p 比 d的规模缩小一半
- 计算$m_p = c^{dp}$ （mod p），计算量为（*） 的1/8
- 计算$m_q = c^{dq}$ （mod q）
- 由 $m=m_p$ （mod p）和$m=m_q$ （mod q）计算 m （mod n），计算量为（*） 的1/4

RSA的安全性

•RSA的安全性是基于分解大整数的困难性假定，如果RSA的模数n被成功地分解为p×q，则立即获得Φ(n)=(p-1)(q-1)，从而能够确定e模Φ(n)的乘法逆元d，即$d \equiv e^{-1} \bmod \Phi(n)$，因此攻击成功。

RSA算法安全性

- 某人一年内车祸死亡的概率　　1万(2^{14})分之一
- 每天被闪电杀死的概率　　90亿(2^{33})分之一
- 彩民赢得双色球彩票头奖概率　4百万(2^{22})分之一
- 赢头奖且同一天被闪电击死的可能性　$1/2^{55}$
- 宇宙中的原子数　　10^{77}(2^{265})
- RSA暴力破解分析次数：　2^{512}　(2^{256})，　2^{768}　(2^{384})，2^{1024}　(2^{512})

大整数分解现状

- 随着人类计算能力的不断提高，原来被认为是不可能分解的大数已被成功分解。

表 1　RSA 模数分解情况

RSA number	decimal digits	binary digits	factored on	factored by
RSA-150	150	496	Apr-04	Kazumaro Aoki
RSA-170	170	563	Dec-09	Bonenberger, *et al*[4]
RSA-180	180	596	May-10	Danilov, *et al*[5]
RSA-190	190	629	Nov-10	Timofeev[6]
RSA-200	200	663	Mar-05	Jens Franke[7]
RSA-210	210	696	Sep-13	Propper, *et al*[8]
RSA-704	212	704	Jul-12	Bai Shi, *et al*[9]
RSA-768	232	768	Dec-09	Kleinjung, *et al*[10]

➢ **模数长度应该介于1024bit到2048bit之间**

n与φ(n)

是否有不通过分解大整数的其它攻击途径？下面我们证明由直接确定 $\varphi(n)$ 等价于对 n 的分解。

设 $n = p \times q$ 中，$p > q$，由 $\varphi(n) = (p-1)(q-1)$，我们有

$$p + q = n - \varphi(n) + 1$$

以及

$$p - q = \sqrt{(p+q)^2 - 4n} = \sqrt{[n - \varphi(n) + 1]^2 - 4n}$$

由此可得，

$$p = \frac{1}{2}\left[(p+q) + (p-q)\right]$$

$$q = \frac{1}{2}\left[(p+q) - (p-q)\right]$$

所以，由 p 、q 确定 $\varphi(n)$ 和由 $\varphi(n)$ 确定 p 、q 是等价的。

为保证算法的安全性，还对 p 和 q 提出以下要求：

（1）$|p-q|$ 要大；

由 $\frac{(p+q)^2}{4}-n=\frac{(p+q)^2}{4}-pq=\frac{(p-q)^2}{4}$，如果 $|p-q|$ 小，则 $\frac{(p-q)^2}{4}$ 也小，因此 $\frac{(p+q)^2}{4}$ 稍大于 n，$\frac{p+q}{2}$ 稍大于 $\sqrt{n}$。

可得 n 的如下分解法：

① 顺序检查大于 $\sqrt{n}$ 的每一整数 x，直到找到一个 x 使得 x^2-n 是某一整数（记为 y）的平方。

② 由 $x^2-n=y^2$，得 $n=(x+y)(x-y)$。

（2）$p-1$ 和 $q-1$ 都应有大素因子；

这是因为RSA算法存在着可能的重复加密攻击法。
设攻击者截获密文 c，可如下进行重复加密：

$$c^e \equiv (m^e)^e \equiv m^{e^2} \pmod{n}$$

$$c^{e^2} \equiv (m^e)^{e^2} \equiv m^{e^3} \pmod{n}$$

$$\cdots$$

$$c^{e^{t-1}} \equiv (m^e)^{e^{t-1}} \equiv m^{e^t} \pmod{n}$$

$$c^{e^t} \equiv (m^e)^{e^t} \equiv m^{e^{t+1}} \pmod{n}$$

若 $m^{e^{t+1}} \equiv c \pmod{n}$，即 $\left(m^{e^t}\right)^e \equiv c \pmod{n}$，则有 $m^{e^t} \equiv m \pmod{n}$，即 $c^{e^{t-1}} \equiv m \pmod{n}$，所以在上述重复加密的倒数第 2 步就已恢复出明文 m，这种攻击法只有在t 较小时才是可行的。

为抵抗这种攻击，p、q 的选取应保证使 t 很大。

设 m 在模 n 下阶为 k，由 $m^{e^t} \equiv m(\bmod n)$ 得 $m^{e^t-1} \equiv 1(\bmod n)$，所以 $k \mid (e^t-1)$，即 $e^t \equiv 1(\bmod k)$，t 取为满足上式的最小值（e 为在模 k 下的阶）。

又当 e 与 k 互素时 $t \mid \varphi(k)$。为使 t 大，k 就应大且 $\varphi(k)$ 应有大的素因子。

又由 $k \mid \varphi(n)$，所以为使 k 大，$p-1$ 和 $q-1$ 都应有大的素因子。

此外，研究结果表明，如果 $e<n$ 且 $d<n^{1/4}$，则 d 能被容易地确定。

对RSA的攻击

- 1. 共模攻击

- 设两个用户的公开钥分别为e_1和e_2，且e_1和e_2互素，明文消息是m，密文分别是

$$c_1 = m^{e_1} \bmod n$$

$$c_2 = m^{e_2} \bmod n$$

- 敌手截获c_1和c_2后，可如下恢复m。用推广的Euclid算法求出满足$re_1+se_2=1$的两个整数r和s，其中一个为负，设为r。再次用推广的Euclid算法求出 c_1^{-1}，由此得

$$c_1^r c_2^s = (c_1^{-1})^{-r} c_2^s \equiv m \bmod n$$

对RSA的攻击（续）

•2. 低指数攻击

• 将RSA同时用于多个用户，然而每个用户的加密指数都很小。设3个用户的模数分别为$n_i(i=1,2,3)$，当$i \neq j$时，$\gcd(n_i, n_j)=1$，否则通过$\gcd(n_i, n_j)$有可能得出n_i 和n_j 的分解。设明文消息是m，密文分别是

•$c_1 \equiv m^3 \pmod{n_1}$

•$c_2 \equiv m^3 \pmod{n_2}$

•$c_3 \equiv m^3 \pmod{n_3}$

•由中国剩余定理可求出$m^3 \pmod{n_1\ n_2\ n_3}$。

•由于$m^3 < n_1\ n_2\ n_3$，可直接由m^3开立方根得到m。

3. 选择密文攻击

RSA是确定性的加密算法, 不能抵御选择密文攻击。

假设攻击者得到了RSA公钥(n,e)，截获到某个密文c_1，假设其明文为m_1,即

$$c_1 = m_1^e \pmod n$$

然后，攻击者选取明文m，构造新密文c_2：

$$c_2 = c_1 m^e \pmod n$$

攻击者让解密者对c_2解密，获得明文m_2，则计算：

$$m_1 = m_2 m^{-1} \pmod n$$

4 乘法同态

- $C_1 = m_1^e \bmod n$
- $C_2 = m_2^e \bmod n$
- $C_1 C_2 = (m_1 m_2)^e \bmod n$

RSA密码体制的知识要点

1. RSA算法步骤
2. 加解密的计算简化
3. p, q的选取
4. 攻击方法
5. 安全性基于大数分解的困难性

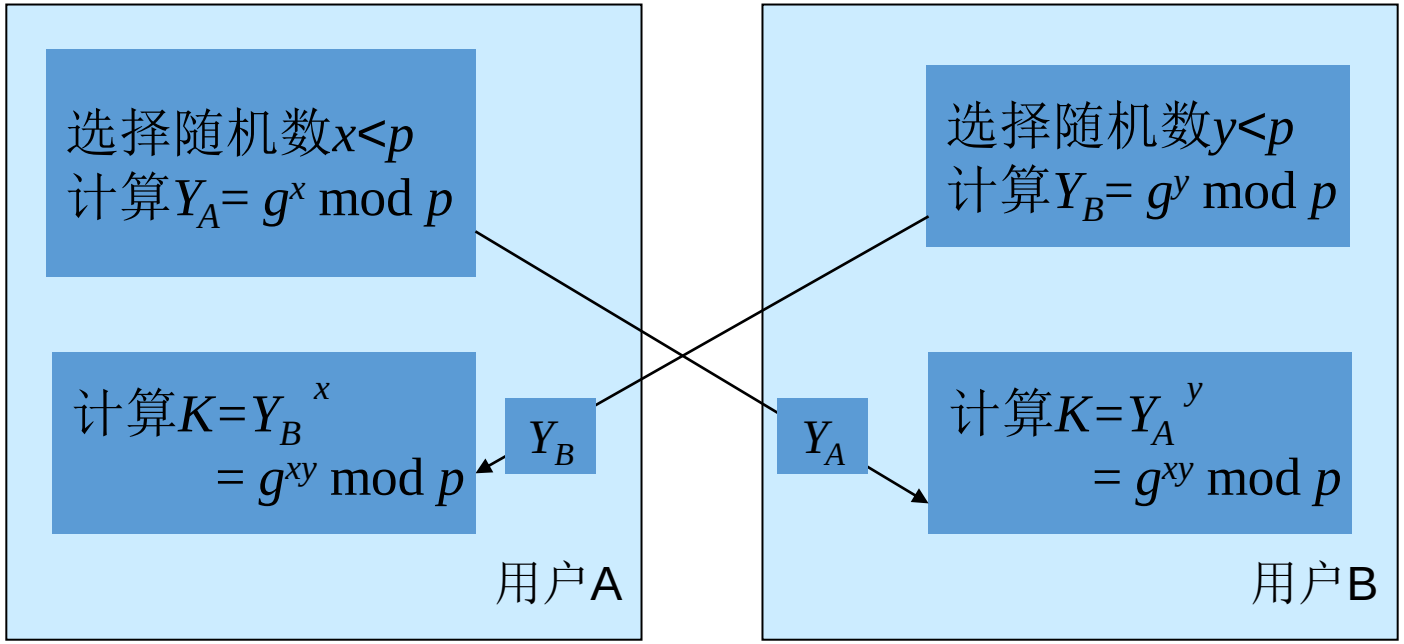
基于离散对数的公钥加密基本思想

- Diffie-Hellman密钥共享

Diffie-Hellman密钥交换

- W.Diffie和M.Hellman1976年提出
- 算法的安全性基于求离散对数的困难性

p是大素数，g是p的本原根，p和g公开



Diffie-Hellman密钥交换

新的问题---计算Diffie-Hellman问题

- Diffie-Hellman密钥交换产生了一个新的问题：
- 已知：g^x，g^y，求：　g^{xy}
- 简写为CDH问题。
- 这个问题衍生出另外一个问题：判定Diffie-Hellman问题，简写为DDH问题，

  即：已知：g^x,g^y，g^z,判定$g^z = g^{xy}$是否成立

问题：

- 是否可以使用Diffie-Hellman协议设计一个公钥加密方案？

ElGamal密码体制

1. 密钥产生过程： 选择一素数p以及小于p的随机数x， g是p的原根，计算 $y \equiv g^x \bmod p$。

   (y，g，p)作为公开密钥，x作为秘密密钥。

2. 加密过程： 明文消息M，随机选一整数 k<p-1，计算

   $C_1 \equiv g^k \ (\bmod\ p)$， $C_2 \equiv y^k M \ (\bmod\ p)$

密文为$C=(C_1, C_2)$。

3. 解密过程：

$$M = c_2 (c_1^x)^{-1} (\bmod\ n) \bmod p$$

$$c_2 (c_1^x)^{-1} (\bmod\ p) = y^k M (g^{kx})^{-1} (\bmod\ p) = y^k M (y^k)^{-1} (\bmod\ p) = M (\bmod\ p)$$

ElGamal密码体制的特点

1. 安全性基于有限域上的离散对数的难解性（不是等价）

   其单向性等价于什么？

2. 加密算法是概率算法

3. 不能抵御选择密文攻击（作业）

4. 存在密文扩张

ElGamal密码的安全性

- 参数要求：p应为150位以上十进制数，500位以上的二进制数，p-1应有大素数因子。
- K必须保密而且必须是一次性的。
  - K泄露，则敌手可计算出y^k从而可以计算出M
  - 使用同一k加密不同的明文M，M’，如果敌手知道M就可由C_2/C_2’=M/M’求出M’

椭圆曲线密码体制

为保证RSA算法的安全性，它的密钥长度需一再增大，使得运算负担越来越大。相比之下，椭圆曲线密码体制ECC（elliptic curve cryptography）可用短得多的密钥获得同样的安全性，因此具有广泛的应用前景。

ECC和RSA/DSA体制在保持同等安全的条件下各自所需的密钥的长度

RSA/DSA	512	768	1024	2048	21000
ECC	106	132	160	211	600

有限域上的椭圆曲线

密码中普遍采用的是有限域上的椭圆曲线，是指曲线方程定义式中，所有系数都是某一有限域GF(p)中的元素（p为一大素数）。其中最为常用的是由方程

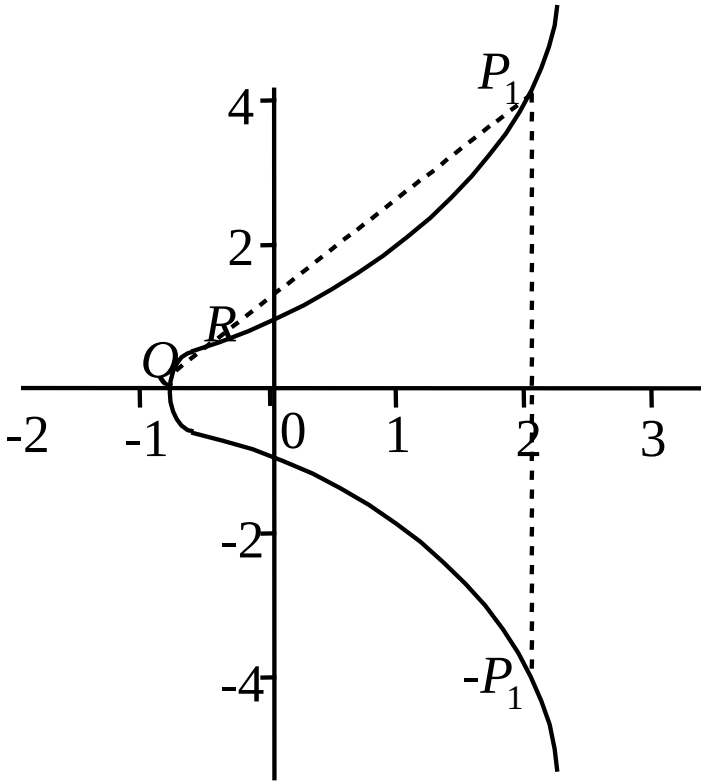
$$y^2 \equiv x^3+ax+b \pmod{p}$$

$$(a,b \in GF(p), 4a^3+27b^2 \neq 0 \pmod{p}) \qquad (4.2)$$

定义的曲线。

例 p=23，a=b=1，$4a^3+27b^2 \pmod{23} \equiv 8 \neq 0$ ，方程(4.2)为$y^2 \equiv x^3+x+1$，其图形是连续曲线。

$$E_p(a,b)=\{(x,y)\mid y^2 \equiv x^3+x+1 \pmod p, 0 \le x,y<p, x,y \in Z_p\} \cup \{O\}$$



点集$E_{23}(1,1)$

(0,1)	(0,22)	(1,7)	(1,16)	(3,10)
(3,13)	(4,0)	(5,4)	(5,19)	(6,4)
(6,19)	(7,11)	(7,12)	(9,7)	(9,16)
(11,3)	(11,20)	(12,4)	(12,19)	(13,7)
(13,16)	(17,3)	(17,20)	(18,3)	(18,20)
(19,5)	(19,18)			

(b)$y^2=x^3+x+1$

一般来说，$E_p(a,b)$由以下方式产生：

① 对每一$x(0 \le x < p$且x为整数），计算$x^3+ax+b(\mathrm{mod}\ p)$。

② 决定①中求得的值在模p下是否有平方根，如果没有，则曲线上没有与这一x相对应的点；如果有，则求出两个平方根（$y=0$ 时只有一个平方根）。

$E_p(a,b)$上的加法定义如下：

设$P, Q \in E_p(a,b)$，则

① $P+O=P$。

② 如果$P=(x,y)$，那么$(x, y)+(x, -y)=O$，即 $(x, -y)$是P的加法逆元，表示为$-P$。

由$E_p(a,b)$的产生方式知，$-P$也是$E_p(a,b)$中的点，如上例，$P=(13,7) \in E_{23}(1,1)$，$-P=(13, -7)$，

而 $-7 \bmod 23 \equiv 16$，所以$-P=(13, 16)$，也在$E_{23}(1,1)$中。

③ 设P=(x_1,y_1)，Q=(x_2,y_2)，P≠-Q，则P+Q=(x_3,y_3)由以下规则确定：

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

其中

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, P = Q \end{cases}$$

椭圆曲线上的密码

椭圆曲线上的数学困难问题

在椭圆曲线构成的Abel群$E_p(a,b)$：

$P \in E_p(a,b)$, P的阶是一个非常大的素数，P的阶是满足$nP=O$的最小正整数n。

$Q=kP$，

(1) 已知k和P易求Q;

(2) 已知P、Q求k则是困难的

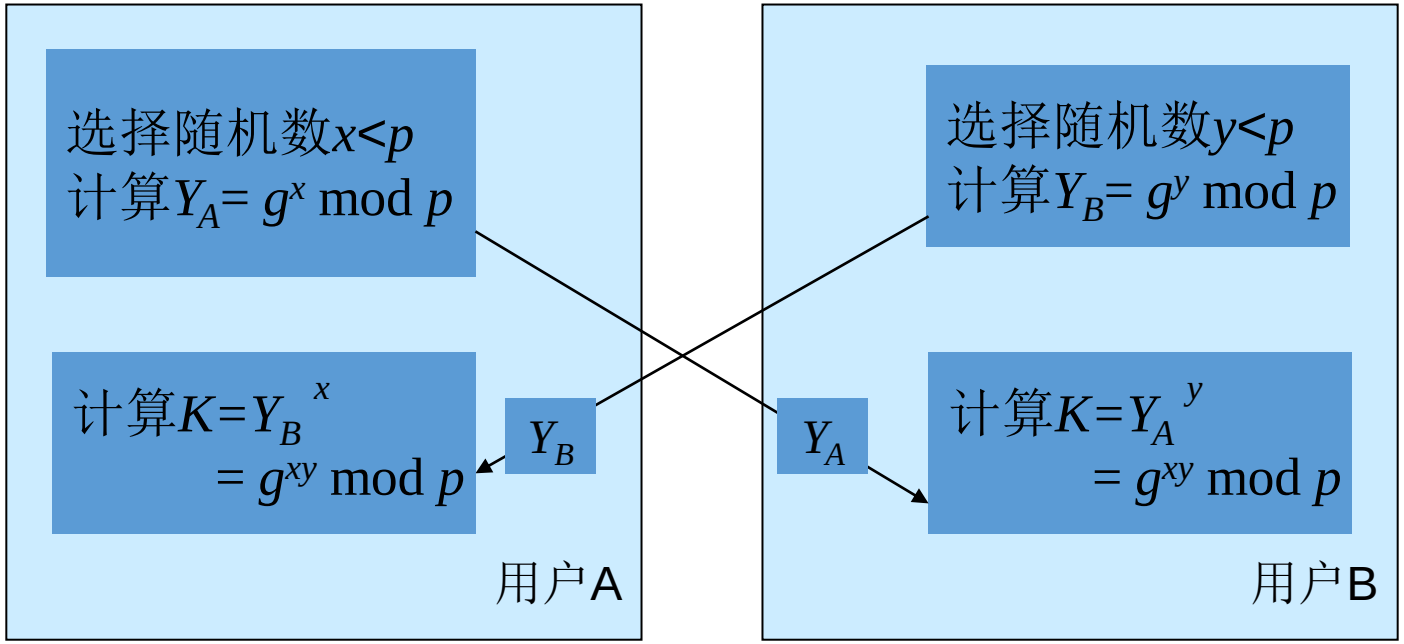
这就是椭圆曲线上的离散对数问题，可应用于构造公钥密码体制。

Diffie-Hellman密钥交换和ElGamal密码体制可推广到椭圆曲线来实现。

Diffie-Hellman密钥交换

- W.Diffie和M.Hellman1976年提出
- 算法的安全性基于求离散对数的困难性

p是大素数，g是p的本原根，p和g公开



Diffie-Hellman密钥交换

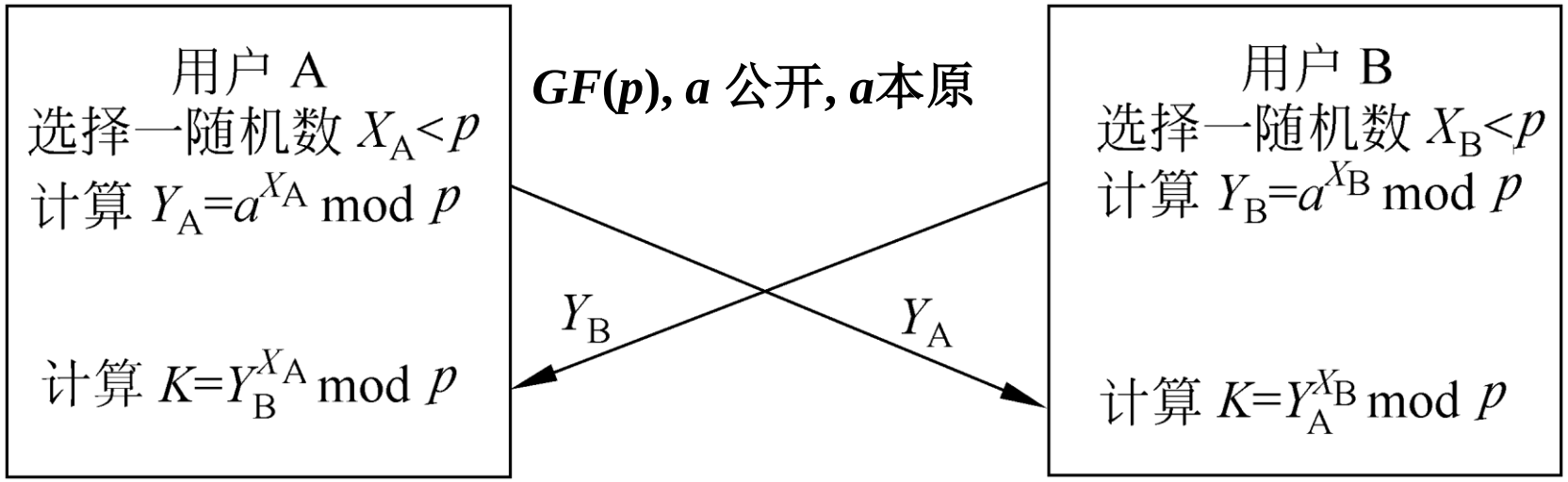
椭圆曲线上的Diffie-Hellman密钥交换

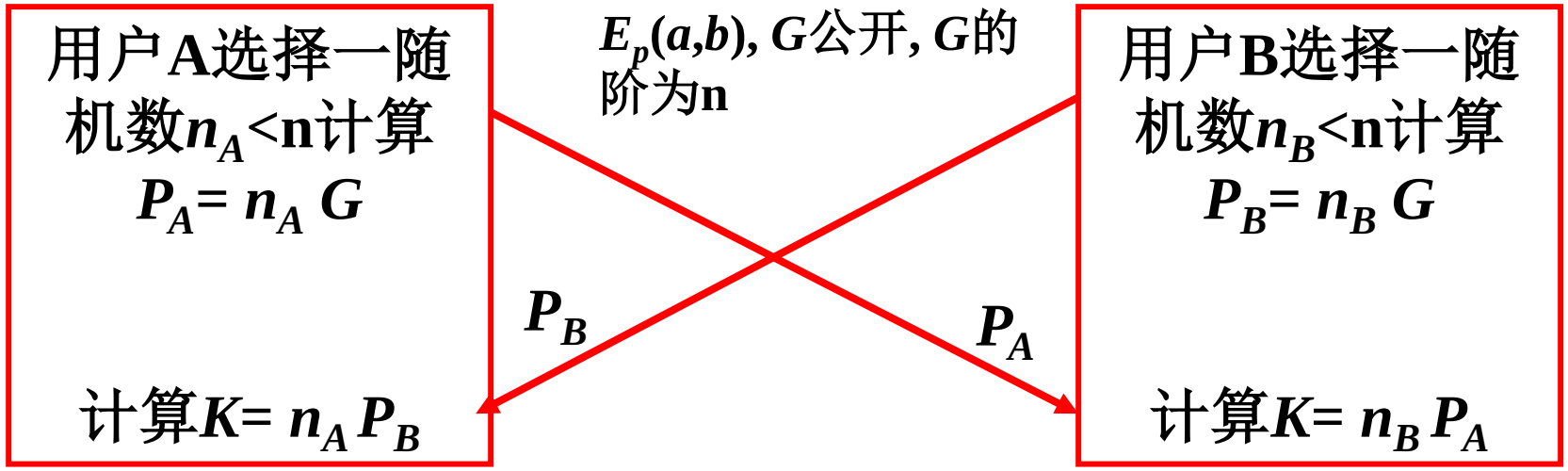
移植到椭圆曲线上：

首先取一素数$p \approx 2^{180}$和两个参数a、b，则得方程(4.2)表达的椭圆曲线及其上面的点构成的Abel群$E_p(a,b)$。

第2步，取$E_p(a,b)$的一个生成元$G(x_1,y_1)$，要求G的阶是一个非常大的素数，G的阶是满足$nG=O$的最小正整数n。

$E_p(a,b)$和G作为公开参数。


用户 A
选择一随机数 $X_A<p$
计算 $Y_A=a^{X_A} \bmod p$
计算 $K=Y_B^{X_A} \bmod p$
$GF(p)$, a 公开, a本原
Y_B
Y_A
用户 B
选择一随机数 $X_B<p$
计算 $Y_B=a^{X_B} \bmod p$
计算 $K=Y_A^{X_B} \bmod p$


用户A选择一随机数n_A<n计算
$P_A= n_A G$
计算$K= n_A P_B$
$E_p(a,b)$, G公开, G的阶为n
P_B
P_A
用户B选择一随机数n_B<n计算
$P_B= n_B G$
计算$K= n_B P_A$

椭圆曲线实现ElGamal密码体制

1. p>3大素数，Z_p，E。G∈E, G的阶足够大。p,E,G公开。

2. 随机选取整数x_A，$1 \le x_A \le ord(G)-1$, 计算$P_A = x_A G$作为公开的加密密钥，x_A是保密的解密密钥。

3. 明文空间为 E,密文空间为$E \times E$

4. B向A发送消息P_m，可选取一随机正整数$1 \le k \le ord(G)-1$，产生密文：

$$C_m=\{kG, P_m+kP_A\}$$

$C_1 \equiv g^k \bmod p$,

$C_2 \equiv y^k M \bmod p$

5. A解密：

$$P_m+kP_A-x_AkG=P_m+k(x_AG)-x_AkG=P_m$$

$$M = c_2(c_1^x)^{-1} (\mathrm{mod}\ n)$$

明文消息嵌入到椭圆曲线上

设明文消息为**m, k**是一个足够大的整数，使得将明文消息镶嵌到椭圆曲线上时，错误概率是2^{-k}。如取**k=30**，对明文**m**，如下计算一系列**x**:

$$x = \{mk + j,\ j = 0,1,2,\cdots\cdots\}$$

$$= \{30m, 30m+1, 30m+2,\cdots\cdots\}$$

直到$x^3+ax+b(\mathrm{mod}p)$有平方根，则得到点$(x, \sqrt{x^3+ax+b})$

反过来，从椭圆曲线点**(x,y)**得到明文消息**m**,只需求出

$$m = \left\lfloor \frac{x}{30} \right\rfloor$$

ECC标准加密算法

- 参数选择

$GF(p)$上椭圆曲线密码基础参数

$$T=<p, a, b, G, n, h>$$

- p是一个大素数，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：

$$y^2 = x^3 + ax + b \ , \ a, b \in GF(p)$$

- E为全体解点和无穷远点组成的群，E_1是其子群。
- 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;
- $h=|E|/n$，并称为余因子，h将交换群E和循环子群E_1联系起来。

ECC密钥生成

$GF(p)$上的椭圆曲线密码 (ElGamal型)

(1)密钥生成

- 用户选择一个随机数d作为私钥，

  $d \in \{1,2,\cdots,n-1\}$。

- 用户计算

  $Q=dG$

  以Q点为自己的公开钥。

- **$GF(p)$上的ElGamal密码**

(1) 密钥生成

- 用户随机地选择一个整数d作为自己保密的解密钥，

  $2 \leqslant d \leqslant p-2$ 。

- 用户计算

  $y = a^d \bmod p$，

  以y为自己的公开钥。

ECC加密

(2)加密：

- 设明文数据为M，$0 \leqslant M \leqslant n-1$。
- 加密过程：
  - ① 选择一个随机数k，$k \in \{1,2,\cdots,n-1\}$。
  - ② 计算点$X_1(x_1, y_1) = kG$。
  - ③ 计算点$X_2(x_2, y_2) = kQ$，如果分量$x_2=0$，则转①。
  - ④ 计算密文$C=Mx_2 \bmod n$。
  - ⑤ 以(X_1, C)为最终密文。

(2) 加密

- 设明文消息M（$0 \leqslant M \leqslant p-1$）
- 加密过程：

① 随机地选取一个整数k，$2 \leqslant k \leqslant p-2$。

②计算：$C_1 = a^k \bmod p$；

$U = y^k \bmod p$；

$C_2 = UM \bmod p$；

③取$C=(C_1, C_2)$为最终密文。

问题

- 加密过程是否有不合理得地方

ECC解密算法

(3)解密：

① 用私钥d求出点X_2：

$$dX_1 = d(kG)$$
$$= k(dG)$$
$$= kQ$$
$$= X_2(x_2, y_2)$$

② 对C 解密：利用x_2计算得到明文

$$M = C\,x_2^{-1} \bmod n$$ 。

(3) 解密

①计算$V = C_1^{\,d} \bmod p$

$$= (\alpha^k)^d \bmod p$$
$$= (\alpha^d)^k \bmod p$$
$$= (y)^k \bmod p$$
$$= U$$

②计算

$$M = C_2\,V^{-1} \bmod p$$

获得明文。

椭圆曲线密码体制的优点

(1) 安全性高： 攻击有限域上的离散对数可用指数积分法，运算复杂度为 $O(\exp\sqrt[3]{(\log p)(\log\log p)^2}$。 对ECC上离散对数攻击并不有效。

攻击ECC上离散对数问题的方法只有大步小步法，复杂度为$O(\exp(\log\sqrt{p_{\max}}))$。$p_{max}$是ECC形成的交换群的阶的最大素因子，因此ECC上的密码体制比基于有限域上离散对数问题的公钥体制更安全

(2) 密钥量小

由攻击两者的算法复杂度可知，在实现相同的安全性能条件下，椭圆曲线密码体制所需的密钥量远比基于有限域上的离散对数问题的公钥体制的密钥量小。

(3) 灵活性好

有限域GF(q)一定的情况下，其上的循环群就定了。而GF(q)上的椭圆曲线可以通过改变曲线参数，得到不同的曲线，形成不同的循环群。因此，椭圆曲线具有丰富的群结构和多选择性。

感 谢 聆 听!

liaoyj@uestc.edu.cn