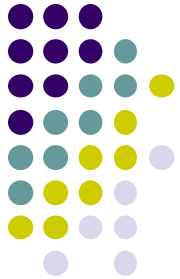
实验课（16学时，8次）

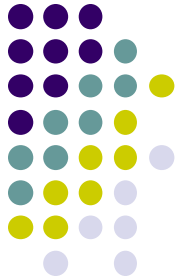
- 实验课地点：
  - 理工学院楼610室
- 按名单序号分三组（人数分别为：30，30，29）
- 时间（从本周开始）
  - 第一组：周一下午第7，8节
  - 第二组：周一晚第9，10节
  - 第三组：周三晚第9，10节



第3章

公钥密码体系与密钥管理

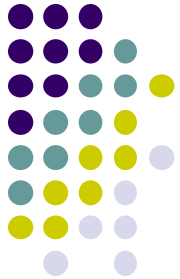
密钥管理



Problem: n users. Storing mutual secret keys is difficult

Total: O(n) keys per user

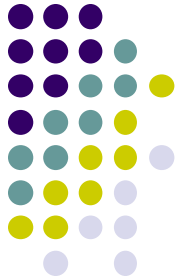
为什么需要公钥密码体系?



- 在网络通信中，使用对称密码算法加密数据时，通信各方必须首先协商好所使用的秘密密钥
  - 通过专人传送秘密密钥
  - 通过开会的方式确定秘密密钥
  - 使用邮寄、E-mail以及电话等方式传送秘密密钥
  - …
  - 但是，这些传统的方式不能适应现代网络通信的需要
- 公钥密码体系(PKC)
  - 于1970年代出现
  - **能够在没有预先共享秘密密钥的情况下进行安全的密钥分发**
  - 用于**身份鉴别**

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

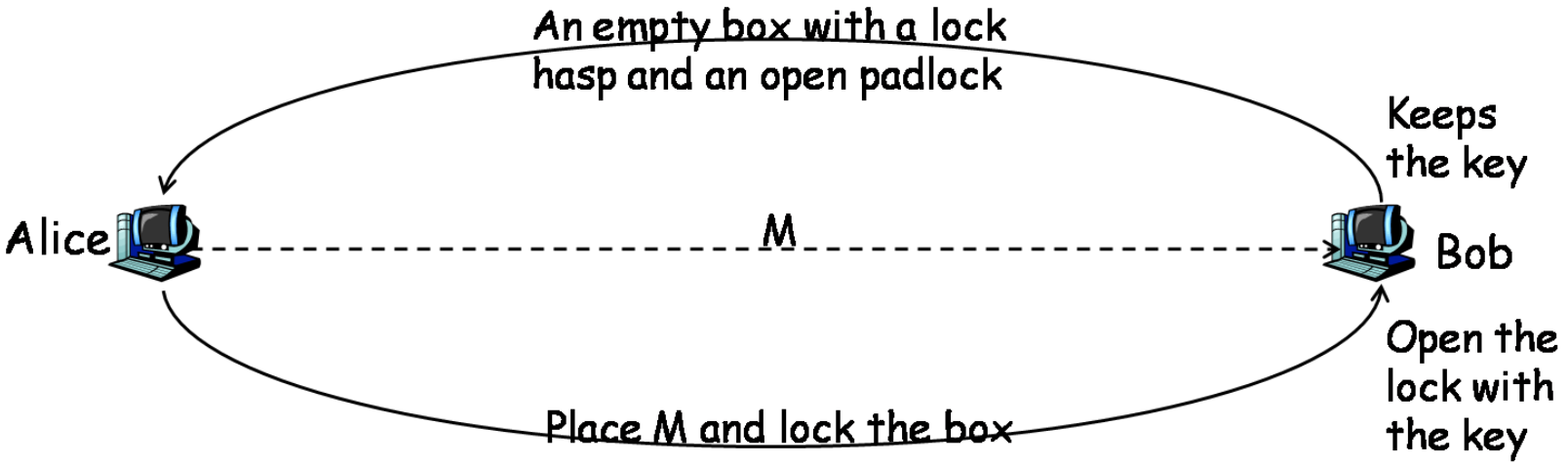
第3章 内容概要



- 3.1 公钥密码体系的定义
- 3.2 数论的基本概念和定理
- 3.3 Diffie-Hellman密钥交换
- 3.4 RSA 密码体系
- 3.5 椭圆曲线密码体系
- 3.6 密钥分配与管理

公钥密码体系的基本思想

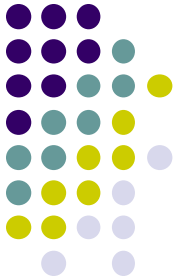
- 采用传统的邮寄方式，Bob能够在没有预先共享秘密的情况下，接收Alice发送的机密消息



- 打开的锁和盒子: 公钥(对外公开)
- Bob保管的钥匙: 私钥(需要保密)
- 问题: 如何用数学的方法实现这个思想?

《计算机网络安全的理论与实践（第2版）》. 【美】王杰，高等教育出版社，2011年.

另一个例子



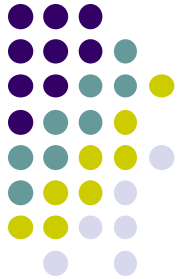
- 假设存在两个函数f_0和f_1，满足$f_1(f_0(a, y), x) = f_1(f_0(a, x), y)$，而且，通过已知的$f_0(a, x)$ 和a很难计算出x
- Alice进行以下步骤：
  - 随机选取一个正整数x_1 (私钥)，然后发送$y_1 = f_0(a, x_1)$ 给Bob
- 类似的，Bob 进行以下步骤：
  - 随机选取一个正整数x_2 (私钥)，然后发送 $y_2 = f_0(a, x_2)$ 给Alice
- Bob 计算 $K_2 = f_1(y_1, x_2)$，Alice 计算 $K_1 = f_1(y_2, x_1)$，作为他们的秘密密钥
- 因为 $f_1(y_2, x_1) = f_1(f_0(a, x_2), x_1) = f_1(f_0(a, x_1), x_2) = f_1(y_1, x_2)$，因此有 $K_1 = K_2$
- Malice 可以在通信线路上监听到y_1 and y_2, 但是不能获得x_1 or x_2
- 问题: 如何找到这样的函数f_1和f_2?

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

公钥密码体系的准则

- 正向计算易解性
  - 计算加密和解密是容易的
  - 产生一对密钥对(K^u, K^r) 是容易的，其中 K^u 是公钥，K^r 是对应的私钥
- 反向计算难解性
  - 通过密文 C 和公钥 K^u 计算明文M是计算难解的
  - 公钥K^u 不能泄露关于私钥K^r的有用信息
- 可交换性
  - (K^u, K^r) 满足 $$M = D_{K^r}(E_{K^u}(M)) = D_{K^u}(E_{K^r}(M)) = E_{K^u}(D_{K^r}(M)) = E_{K^r}(D_{K^u}(M))$$
  - 对身份验证和数字签名是需要的，但对于密钥交换不是必需的

第3章 内容概要



- 3.1 公钥密码体系的定义
- 3.2 数论的基本概念和定理
- 3.3 Diffie-Hellman密钥交换
- 3.4 RSA 密码体系
- 3.5 椭圆曲线密码体系
- 3.6 密钥分配与管理

- 算数基本定理
  - 任意大于1的整数都可表示为若干个素数的乘积。而且，如果这些素数按非降序排列的话，这种表示是唯一的。

$$n = p_1^{a_1} p_2^{a_2} \cdots p_t^{a_t},$$

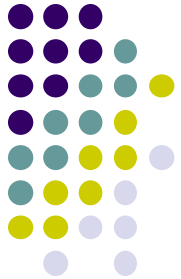
$$p_1 < p_2 < \ldots < p_t \text{ are prime numbers,}$$

$$a_i \ (i = 1, \ldots, t) \text{ is a positive integer}$$

- 素数定理
  - 设 n 是大于1的整数，$\pi(n)$ 表示小于n的素数的个数，则

$$\pi(n) \sim n/\ln n$$

Facts About Numbers



- Prime number p:
  - p is an integer
  - $p \geq 2$
  - The only divisors of p are 1 and p
- Examples
  - $2, 7, 19$ are primes
  - $-3, 0, 1, 6$ are not primes
- Prime decomposition of a positive integer n:

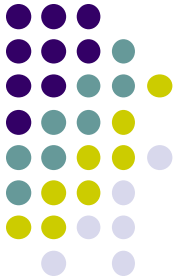
$$n = p_1^{e_1} \times \ldots \times p_k^{e_k}$$

- Example:
  - $200 = 2^3 \times 5^2$

Fundamental Theorem of Arithmetic

The prime decomposition of a positive integer is unique

Greatest Common Divisor



- The greatest common divisor (GCD) of two positive integers a and b, denoted $\gcd(a, b)$, is the largest positive integer that divides both a and b
- The above definition is extended to arbitrary integers
- Examples:

  $\gcd(18, 30) = 6$ $\qquad$ $\gcd(0, 20) = 20$

  $\gcd(-21, 49) = 7$

- Two integers a and b are said to be relatively prime if

  $$\gcd(a, b) = 1$$

- Example:
  - Integers 15 and 28 are relatively prime

Modular Arithmetic

- Modulo operator for a positive integer n

  $$r = a \bmod n$$

  equivalent to

  $$a = r + kn$$

  and

  $$r = a - \lfloor a/n \rfloor n$$

- Example:

  $29 \bmod 13 = 3$ $\quad$ $13 \bmod 13 = 0$ $\quad$ $-1 \bmod 13 = 12$

  $29 = 3 + 2\times13$ $\quad$ $13 = 0 + 1\times13$ $\quad$ $12 = -1 + 1\times13$

- Modulo and GCD:

  $$\gcd(a, b) = \gcd(b, a \bmod b)$$

- Example:

  $\gcd(21, 12) = 3$ $\quad$ $\gcd(12, 21 \bmod 12) = \gcd(12, 9) = 3$

Euclid's GCD Algorithm

- Euclid's algorithm for computing the GCD repeatedly applies the formula

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

- Example
  - $\gcd(412, 260) = 4$

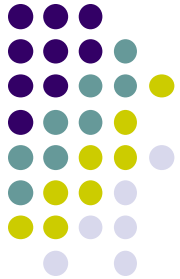
```
Algorithm EuclidGCD(a, b)
    Input integers a and b
    Output gcd(a, b)

    if b = 0
        return a
    else
        return EuclidGCD(b, a mod b)
```

a	412	260	152	108	44	20	4
b	260	152	108	44	20	4	0

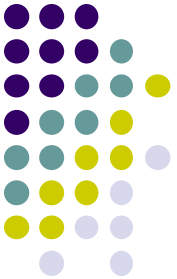
- 模运算
  - 设a 和 b是整数，m 是正整数，则
  - $(a + b) \bmod m = (a \bmod m + b \bmod m) \bmod m$
  - $(a - b) \bmod m = (a \bmod m - b \bmod m) \bmod m$
  - $(a \times b) \bmod m = (a \bmod m \times b \bmod m) \bmod m$
- 同余关系
  - 若$m \mid a - b$，即$a - b$被m整除，则称a和b 在模m下同余, 记作

$$a \equiv b \pmod{m}$$



- 模下的逆元素:
  - 设 a 和 n 是正整数，且$a < n$。若存在正整数 $b < n$， 满足 $a \bullet b \equiv 1 \pmod{n}$, 则称 b 是 a模n的逆
  - 求模逆是RSA密码体制中的一个基本运算
  - 注意：在某些情况下，模逆不一定存在

Multiplicative Inverses【模的乘法的逆】(1)



- The residues modulo a positive integer n are the set

$$Z_n = \{0, 1, 2, \ldots, (n-1)\}$$

- Let x and y be two elements of Z_n such that

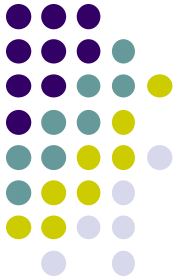
$$xy \bmod n = 1$$

We say that y is the multiplicative inverse of x in Z_n and we write $y = x^{-1}$

- Example:
  - Multiplicative inverses of the residues modulo 11

x	0	1	2	3	4	5	6	7	8	9	10
x^{-1}		1	6	4	3	9	2	8	7	5	10

Multiplicative Inverses (2)



Theorem

An element x of Z_n has a multiplicative inverse if and only if x and n are relatively prime

- Example
  - The elements of Z_{10} with a multiplicative inverse are $1, 3, 7, 9$

Corollary

If is p is prime, every nonzero residue in Z_p has a multiplicative inverse

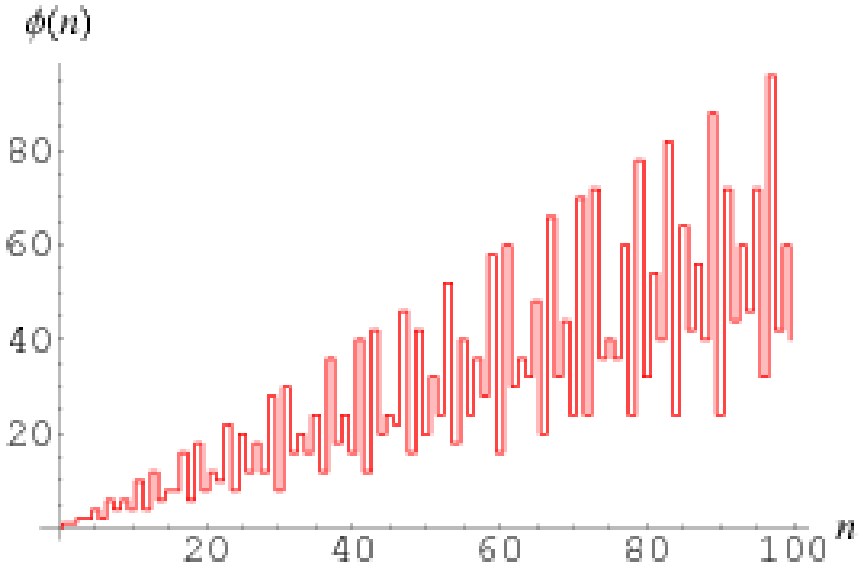
Theorem

A variation of Euclid's GCD algorithm computes the multiplicative inverse of an element x of Z_n or determines that it does not exist

x	0	1	2	3	4	5	6	7	8	9
x^{-1}		1		7				3		9

- 欧拉函数
  - 小于n且与n互素的正整数的个数为

$$\phi(n) = \left[p_1^{\alpha_1 - 1}(p_1 - 1)\right]\left[p_2^{\alpha_2 - 1}(p_2 - 1)\right]\cdots\left[p_t^{\alpha_t - 1}(p_t - 1)\right]$$

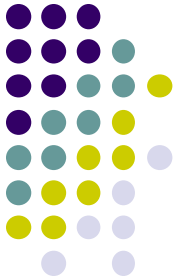


Powers【模的指数幂】

- Let p be a prime
- The sequences of successive powers of the elements of Z_p exhibit repeating subsequences
- The sizes of the repeating subsequences and the number of their repetitions are the divisors of $p - 1$
- Example ($p = 7$)

x	x^2	x^3	x^4	x^5	x^6
1	1	1	1	1	1
2	4	1	2	4	1
3	2	6	4	5	1
4	2	1	4	2	1
5	4	6	2	3	1
6	1	6	1	6	1

Fermat's Little Theorem费马小定理



Theorem

Let p be a prime. For each nonzero residue x of Z_p, we have $x^{p-1} \bmod p = 1$

- Example ($p = 5$):

$1^4 \bmod 5 = 1$ $\quad$ $2^4 \bmod 5 = 16 \bmod 5 = 1$

$3^4 \bmod 5 = 81 \bmod 5 = 1$ $\quad$ $4^4 \bmod 5 = 256 \bmod 5 = 1$

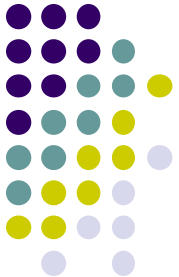
Corollary

Let p be a prime. For each nonzero residue x of Z_p, the multiplicative inverse of x is $x^{p-2} \bmod p$

Proof

$$x(x^{p-2} \bmod p) \bmod p = xx^{p-2} \bmod p = x^{p-1} \bmod p = 1$$

Euler's Theorem欧拉定理



- The multiplicative group for Z_n, denoted with Z^*_n, is the subset of elements of Z_n relatively prime with n
- The totient function of n, denoted with $\phi(n)$, is the size of Z^*_n
- Example

  $Z^*_{10} = \{ 1, 3, 7, 9 \}$ $\quad\quad$ $\phi(10) = 4$
- If p is prime, we have

  $Z^*_p = \{1, 2, \ldots, (p-1)\}$ $\quad\quad$ $\phi(p) = p - 1$

Euler's Theorem

For each element x of Z^*_n, we have $x^{\phi(n)} \bmod n = 1$

- Example ($n = 10$)

  $3^{\phi(10)} \bmod 10 = 3^4 \bmod 10 = 81 \bmod 10 = 1$

  $7^{\phi(10)} \bmod 10 = 7^4 \bmod 10 = 2401 \bmod 10 = 1$

  $9^{\phi(10)} \bmod 10 = 9^4 \bmod 10 = 6561 \bmod 10 = 1$

- 欧拉定理:
  - 设a 是一个正整数，n 是大于1且与a互素的整数, 则

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

- 费马小定理:
  - 设 p 是一个素数，a 是不被p整除的正整数, 则

$$a^{p-1} \equiv 1 \pmod{p}$$

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

- 原根:
  - 设a是正整数，若对于任意的正整数$m < \varphi(n)$,都有 $a^m \not\equiv 1(mod\ n)$ ，则a 称为模n的原根。
  - 不是所有的整数 n 都有原根
- 快速模幂:
  - 计算a^x mod n 是公钥密码学中的常用运算
  - 计算a^x mod n的直接方法: 先计算a^x, 然后计算模 n. 这需要很高的时间复杂度!!!
  - 设x 是一个正整数，令 $k = \lfloor log_2 x \rfloor, b_i \in \{0,1\}$ ，则

$$x = b_k \cdot 2^k + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0 = \sum_{b_i=1} 2^i$$

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

- 快速模幂算法如下：

$$a^x \bmod n = a^{(\sum_{b_i=1} 2^i)} \bmod n = [\prod_{b_i=1} a^{2^i}] \bmod n = [\prod_{b_i=1} a^{2^i} \bmod n] \bmod n$$

fast modular exponentiation algorithm :

1. let $g_k = a$.

2. foreach integer i from k - 1 down to 0

3. let $g_i = (g_{i+1} \times g_{i+1}) \bmod n$;

4. if $b_i = 1$ let $g_i = (g_i \times a) \bmod n$.

g_0 (the output of the algorithm) is equal to $a^x \bmod n$.

- 一个实际的例子见课本的第85页。

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

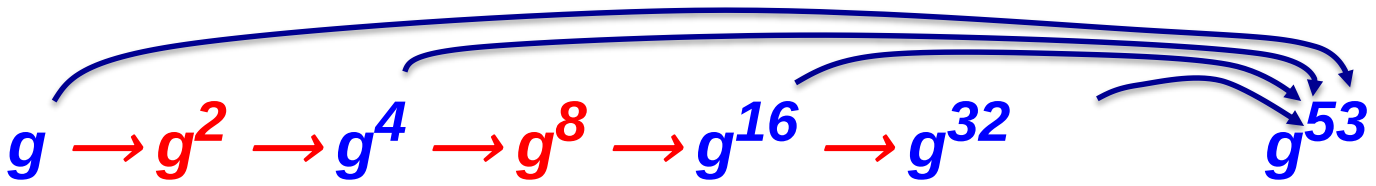
另一种方法计算 Exponentiation

Finite cyclic group G (for example G = $\mathbb{Z}_p^*$)

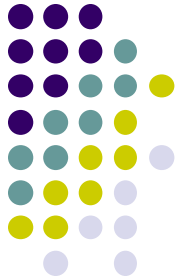
Goal: given g in G and x compute g^x

Example: suppose x = 53 = $(110101)_2$ = 32+16+4+1

Then: $g^{53} = g^{32+16+4+1} = g^{32} \cdot g^{16} \cdot g^4 \cdot g^1$



The repeated squaring alg. (重复平方算法)



Input: g in G and x>0 ; **Output**: g^x

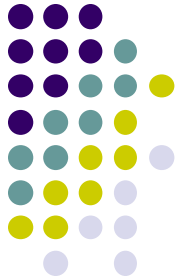
write $x = (x_n\, x_{n-1} \ldots x_2\, x_1\, x_0)_2$

```
y ← g   ,   z ← 1
for i = 0 to n do:
      if  (x[i] == 1):     z ← z·y
      y ← y^2
output  z
```

example: g^{53}

y	z
g^2	g
g^4	g
g^8	g^5
g^{16}	g^5
g^{32}	g^{21}
g^{64}	$\mathbf{g^{53}}$

寻找大素数



- 寻找一个k比特的素数并不难，问题是如何快速判断n是否为素数
- 如何快速地确定一个给定的奇数是否是素数
  - 检查 n 否有一个因子 $1 < x \le \sqrt{n}$
    - 时间复杂性:

$$O(\sqrt{n}) = O(2^{\frac{1}{2}\log n})$$

- Miller-Rabin素数测试算法（基于概率的算法）
  - 这是一个概率算法; 返回错误信息的概率小于 2^{-2m}, 其中 m 是算法的运行次数
  - 设 n 是一个 大于 1的奇数，k 是一个正整数，满足 $n-1=2^k q$, 其中 q 是奇数

1. Choose at random an integer a with $1 < a < n-1$.
2. If $a^q \bmod p \neq 1$ and $a^{2^j q} \bmod p \neq -1$ for all j with $1 \leq j \leq k-1$, then output "n is not prime" and halt.
3. Otherwise, output "n is likely to be prime". Then choose at random another integer a with $1 < a < n-1$, and repeat Step 2.

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

中国余数定理

- 求解同余方程组的方法
- 设 i 是正整数, $Z_i = \{0, \ldots, i\text{-}1\}$
- 设 $n_1, n_2, \ldots, n_k$ 是两两互素的正整数， $n = n_1 \times n_2 \times \ldots \times n_k$
- 对于任意的同余方程组

$$x \equiv a_i (\mathrm{mod}\ n_i), \text{其中 } i = 1, \ldots, k,$$

存在Z_n中的唯一解：

$$x = \left(\sum_{i=1}^{k} a_i b_i\right) \mathrm{mod}\ n$$

其中$b_i = m_i\,(m_i^{-1} \mathrm{mod}\ n_i)$， $m_i = n/n_i$

有限连分数

- 有限连分数是下列形式的分数:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots + \cfrac{1}{a_k}}}}$$

其中 a_0 是整数，$a_1 \ldots, a_k$ 是非零整数

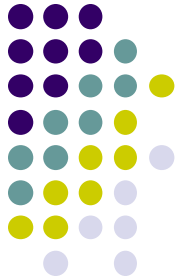
- 给定一个实数x, 我们可以采用下面的算法构造x的连分数:

Construction of Continued Fractions

1. *Set* $x_0 \leftarrow x, a_0 \leftarrow \lfloor x_0 \rfloor, i \leftarrow 0.$
2. *If* $x_i = a_i$, *then halt. Otherwise, set*
3. $$x_{i+1} \leftarrow \frac{1}{x_i - a_i},$$
4. $$a_{i+1} \leftarrow \lfloor x_{i+1} \rfloor$$
5. *Set* $i \leftarrow i + 1$ *and goto Step 2.*

《计算机网络安全的理论与实践（第2版）》. 【美】王杰, 高等教育出版社, 2011年.

第3章 内容概要



- 3.1 公钥密码体系的定义
- 3.2 数论的基本概念和定理
- 3.3 Diffie-Hellman密钥交换
- 3.4 RSA 密码体系
- 3.5 椭圆曲线密码体系
- 3.6 密钥分配与管理

Diffie-Hellman密钥交换

- Diffie and Hellman提出了一种构造两个函数 f_0 和 f_1 的方法:

$$f_0(p, a; x) = a^x \bmod p,$$

$$f_1(x, b) = x^b \bmod p$$

其中 p 是一个大素数，a 为模p的原根;

公开参数: (p, a); 秘密参数: x

$$\begin{aligned} f_1(f_0(p,a;y),x) &= (a^y \bmod p)^x \bmod p \\ &= a^{yx} \bmod p = f_0(p,a;x \cdot y), \\ f_1(f_0(p,a;x),y) &= (a^x \bmod p)^y \bmod p \\ &= a^{xy} \bmod p = f_0(p,a;x \cdot y). \end{aligned}$$

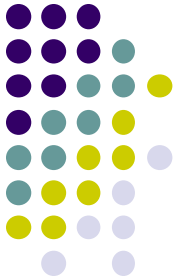
- 因此, $f_1(f_0(p, a; y), x) = f_1(f_0(p, a; x), y)$

《计算机网络安全的理论与实践（第2版）》. 【美】王杰, 高等教育出版社, 2011年.

D-H 密钥交换协议

- **Alice**:
  - 随机选取一个秘密的正整数$X_A < p$
  - 发送 $Y_A = f_0(p, a; X_A) = a^{XA} \bmod p$ 给 Bob，a 和Y_A是公开的
  - 计算 $K_A = f_1(Y_B, X_A) = Y_B^{XA} \bmod p$，作为对称密码算法的秘密密钥, 其中$Y_B$是由Bob发送过来的公开参数
- **类似地，Bob**:
  - 随机选取一个秘密的正整数$X_B < p$
  - 计算并发送$Y_B = f_0(p, a; X_B) = a^{XB} \bmod p$给Alice
  - 计算$K_B = f_1(Y_A, X_B) = Y_A^{XB} \bmod p$

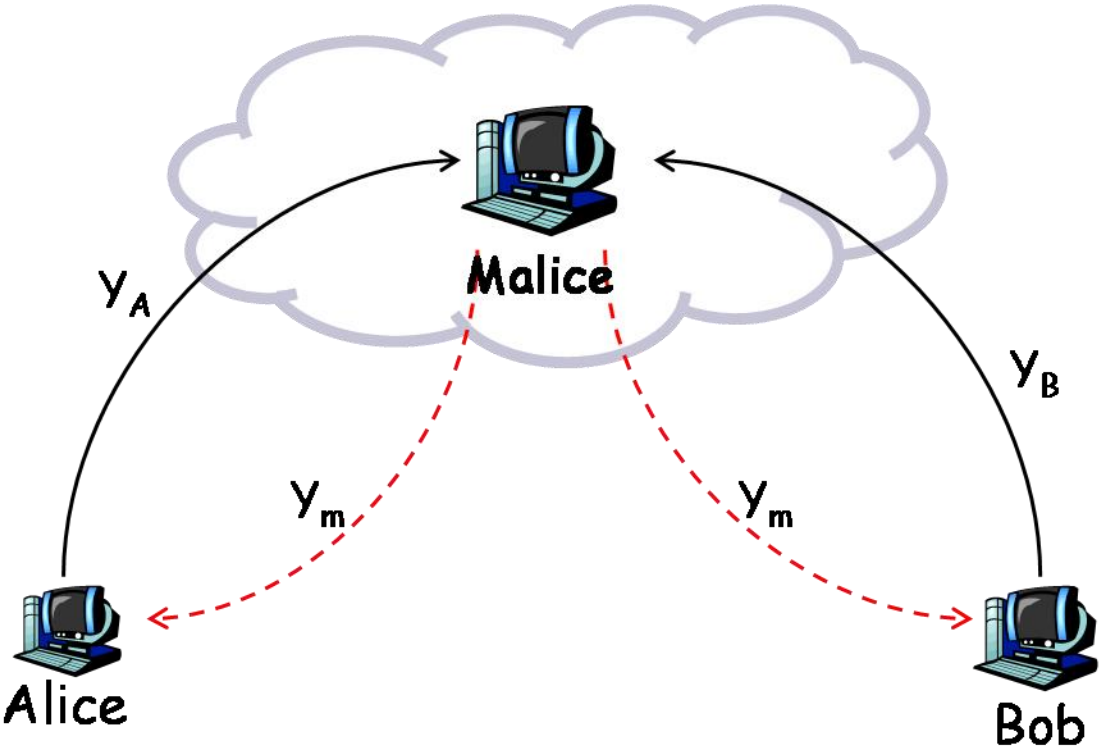
《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.



- Alice和Bob 产生了一个共同的秘密密钥$K = K_A = K_B$
- 正向计算难解性: 快速模幂运算
- 反向难解性: 依赖于计算离散对数问题的困难性，即当$y = a^x \bmod p$，已知y和a, 求解x的困难性
- 目前尚没有求解离散对数问题的多项式时间算法
  - 当素数 p 足够大时, D-H 密钥交换协议是安全的
- Malice 能够监听Y_A 或 Y_B，却不能有效求解出 X_A 或 X_B; 但是，此协议易受到中间人攻击

中间人攻击

- Alice 和 Bob 计算:



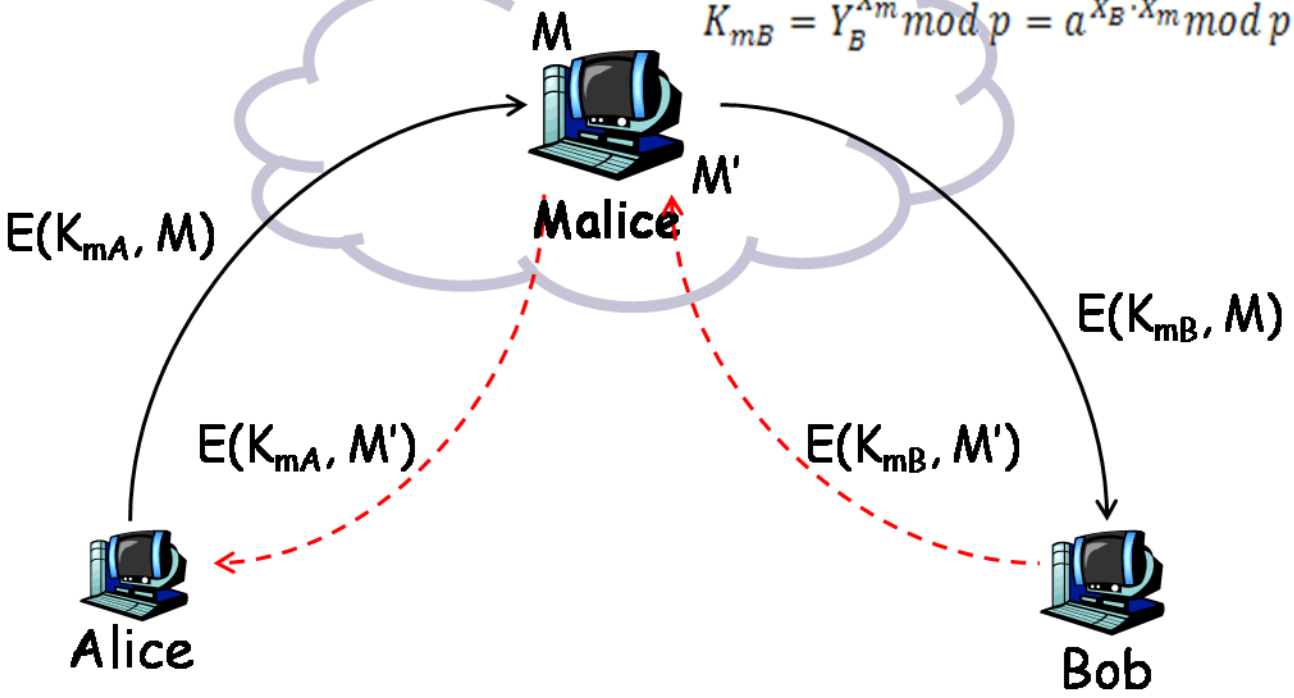
$$K_A = Y_m^{X_A} mod\ p = a^{X_m \cdot X_A} mod\ p \qquad K_B = Y_m^{X_B} mod\ p = a^{X_m \cdot X_B} mod\ p$$

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

- Malice 计算:

$$K_{mA} = Y_A^{X_m} \bmod p = a^{X_A \cdot X_m} \bmod p = K_A$$

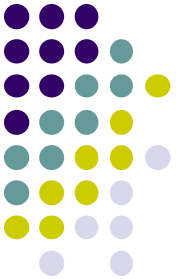
$$K_{mB} = Y_B^{X_m} \bmod p = a^{X_B \cdot X_m} \bmod p = K_B$$



- Alice 与Malice 之间建立了一个共享的秘密密钥
- Bob 与Malice之间建立了一个共享的秘密密钥
- 但是，Alice 和 Bob之间并没有建立共享的秘密密钥

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

Elgamal 公钥密码体系



- 1985年提出，基于Diffie-Hellman困难性假设
- Alice 加密明文 M：

$$1.\ Select\ a\ positive\ integer\ k\ at\ random\ with\ k < p;$$

$$2.\ Compute\ K = (Y_B)^k mod\ p;$$

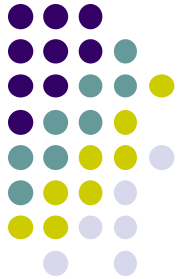
$$3.\ Compute\ C_1 = a^k mod\ p, C_2 = (K \cdot M) mod\ p\ and\ send\ (C_1, C_2)\ to\ Bob.$$

- 接收到密文(C_1, C_2)后, Bob 通过下面的方法进行解密：

$$M = \left(C_2 \cdot \left(C_1^{X_B} mod\ p\right)^{-1}\right) mod\ p$$

《计算机网络安全的理论与实践（第2版）》. 【美】王杰，高等教育出版社，2011年.

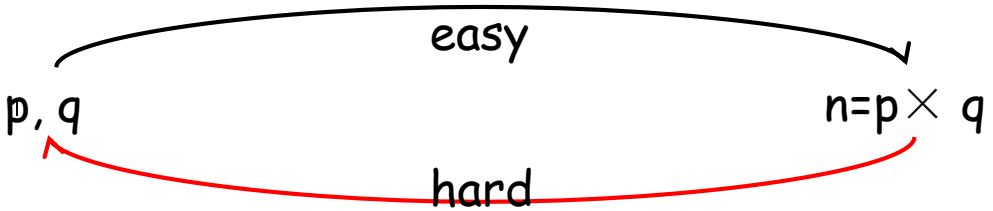
第3章 内容概要



- 3.1 公钥密码体系的定义
- 3.2 数论的基本概念和定理
- 3.3 Diffie-Hellman密钥交换
- 3.4 RSA 密码体系
- 3.5 椭圆曲线密码体系
- 3.6 密钥分配与管理

RSA 密钥, 加密和解密

- 基本运算: 模幂
- 选取大素数p 和q，令$n = p\,q$
- 选取正整数d ，满足 $1 < d < \varphi(n)$，以及$\gcd(d, \varphi(n)) = 1$
- 计算 $e = d^{-1} \bmod \varphi(n)$
- 公钥: (e, n); 私钥: d
- 加密: $C = M^e \bmod n$
- 解密: $M = C^d \bmod n$
- 正向有效性: 快速模幂
- 反向难解性: 整数分解



- 可交换性: 满足

《计算机网络安全的理论与实践（第2版）》. 【美】王杰, 高等教育出版社, 2011年.

RSA参数的攻击

- 尝试所有可能的私钥对密文进行解密
  - 暴力搜索，在多项式时间内是不可行的.
- 分解大整数n
  - 尚未知此问题是否是多项式时间可解的 （试图找到p和q）
- 采用定时分析计算私钥
  - 模幂运算需要的时间与指数的二进制表示中的0和1的个数相关
- 通过泄露的部分秘密信息，计算RSA 的私钥

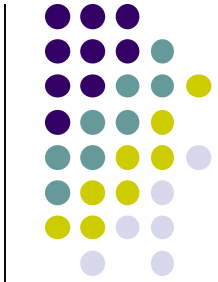
小指数攻击 Attacks

- 设Alice的公钥为 $K_A^u = (e, n_A)$, Bob的公钥为 $K_B^u = (e, n_B)$ ， 且 $\gcd(n_A, n_B) = 1$
- Charlie 发送关于M ($M < \min\{n_A, n_B\}$)的消息给Alice和Bob:
  - $C_A = M^2 \bmod n_A$ to Alice
  - $C_B = M^2 \bmod n_B$ to Bob
- Malice 窃听到C_A 和 C_B，用中国余数定理求解下面的同余方程组:

$$x \equiv C_A (\bmod\ n_A)$$
$$x \equiv C_B (\bmod\ n_B)$$

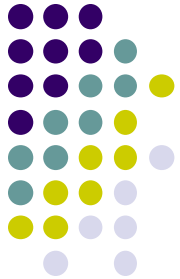
- 设 $x_0 \in Z_n$ 是一个解, 其中$n=n_An_B$， 则有 $x_0 = M^2 \bmod n$. 由于 $M < \sqrt{n}$, 因此有$x_0 = M^2$, $M = \sqrt{x_0}$ 。

部分信息泄露攻击



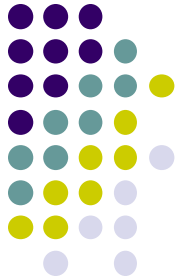
- 设 m 是n 的十进制表示的长度
- 如果素数p (或 q)的低$m/4$ 比特泄露了, 则 n 能够被有效分解
- 如果d的后m/4位数字泄露，则也可快速算出d
- 当 d 已经不安全时，不应继续用原先的秘密参数p,q去产生新的公钥和私钥；因为用原来的p和q 生成一对新的 d 和 e ，也会有利于分解n

其它攻击方法



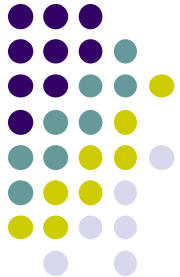
- 明文M 不应是素数p 或 q的倍数
  - 否则n 能够被有效分解
- 如果明文 M是短的消息，并且是两个近似长度的整数的乘积, 则 Malice 能够用中间人攻击的方法计算 M:
  - $M = m_1 \cdot m_2$ ， $|M| = l$
  - Malice 截获到密文 $C = M^e \bmod n$, 计算并排序以下的数组:
    - 对每一个正整数$x \le 2^{l/2+1}$, 计算 $Cx^{-e} \pmod n$
    - 对每一个正整数$y \le 2^{l/2+1}$, 计算 $y^e \pmod n$
  - 如果存在整数x 和 y ，满足$Cx^{-e} \pmod n = y^e \pmod n$, 则 $C \equiv (xy)^e \pmod n$. 因此, $M \equiv C^{-e} \equiv xy \pmod n$
  - 时间复杂性: $O(2^{l/2+2})$，比穷举的量级小
  - 抵御方法: 明文前后加入一些无用字符使新明文不等于两个近似长度的整数乘积

第3章 内容概要



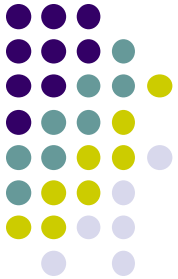
- 3.1 公钥密码体系的定义
- 3.2 数论的基本概念和定理
- 3.3 Diffie-Hellman密钥交换
- 3.4 RSA 密码体系
- 3.5 椭圆曲线密码体系
- 3.6 密钥分配与管理

密钥分发与管理



- 公钥密码体系比传统的对称密码体系更耗时
  - 公钥密码体系不适合用于加密大规模的数据
- 公钥密码体系常用于加密秘密密钥，或用于加密较短的认证的消息等

主密钥与会话密钥



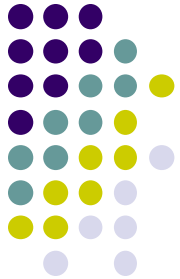
- 主密钥(K_m): 是用于加密其它密钥的密钥
  - 降低主密钥泄露的可能性
- 会话密钥(K_s): 每一次通信会话用的密钥，用主密钥加密保护
  - 用于加密一个消息或TCP数据包
  - 比主密钥的生命周期短

公钥证书

- 在公钥密码体系中, 用户必须知道别人的公钥
  - 在专门的网站发布或Email发送
  - 不能确认公钥的真实拥有者
- 公钥证书用于验证公钥的所有者
  - 由可信组织签发 (CAs)
  - CA 用公钥密码体系 进行证书的验证
    - 在网站上公布他的公钥
    - 为每一个用户签发证书
    - 用私钥对证书进行签名
- 当Alice 需要用 Bob 的公钥时:
  - 让 Bob 发送他的证书
  - 用 CA的公钥验证证书的合法性
  - 从证书中获取Bob的公钥

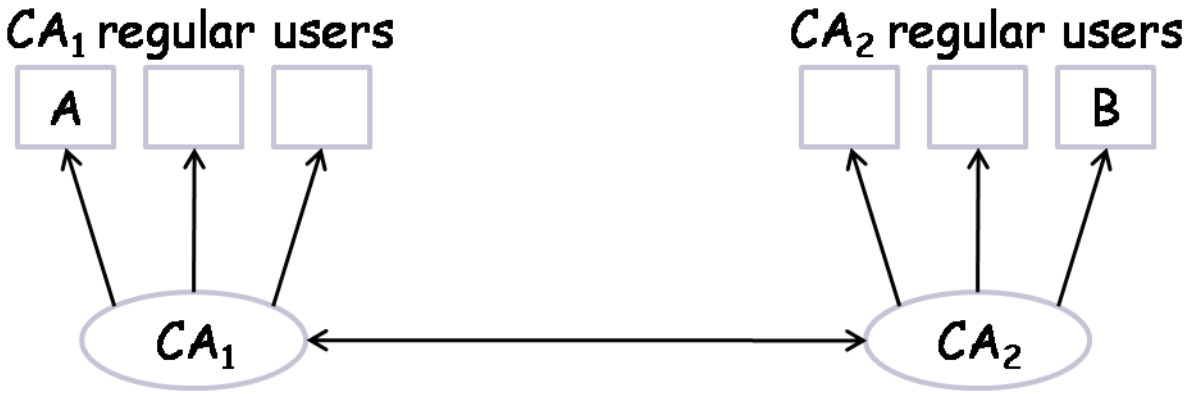
《计算机网络安全的理论与实践（第2版）》. 【美】王杰, 高等教育出版社, 2011年.

CA网络



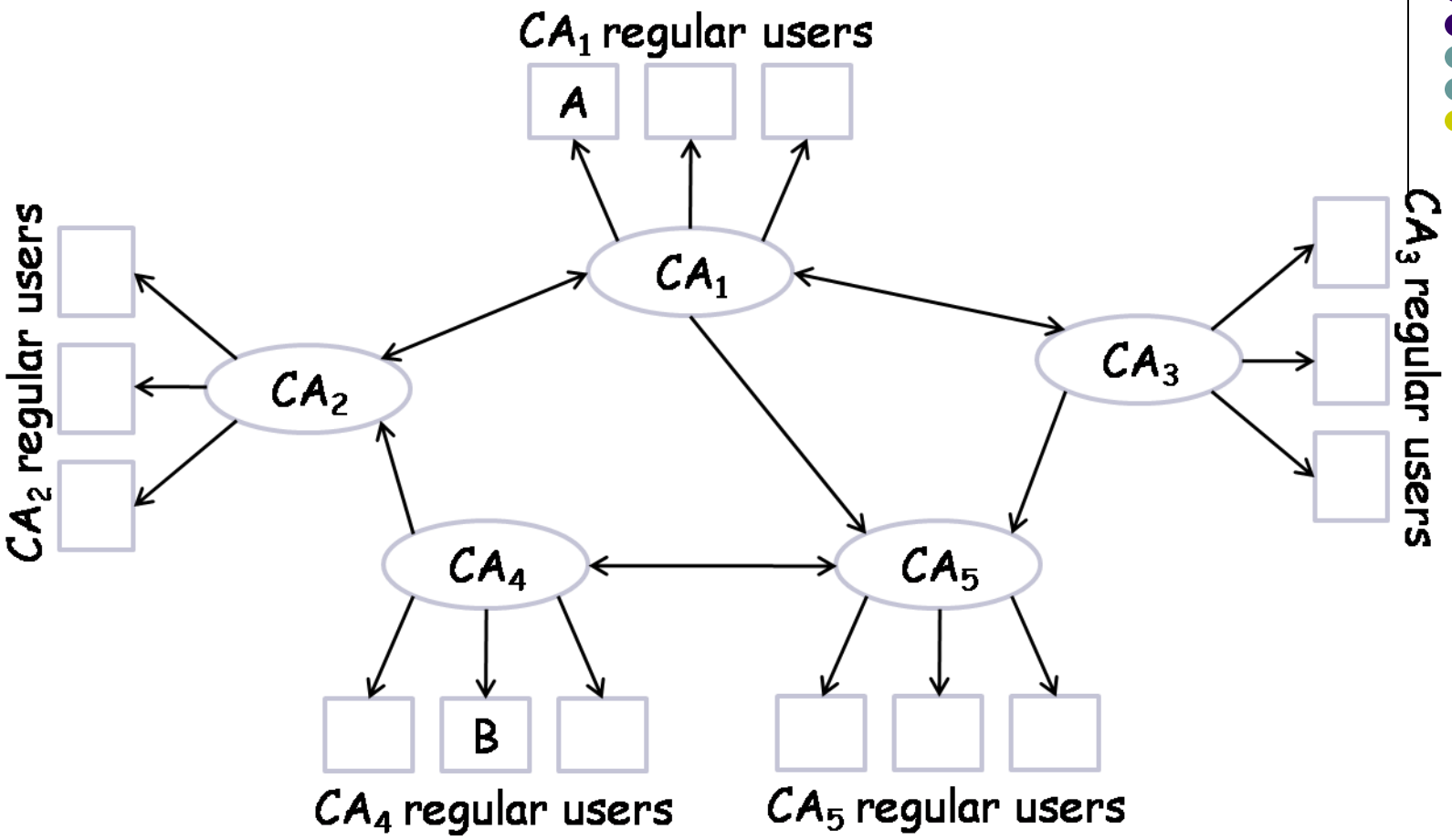
- CA 需要管理证书的签发、过期、撤销等环节
- CA(K_X^u): CA 给用户X 签发的证书，其公钥为 K_X^u
- 如果Alice和Bob 的证书由不同的 CA中心签发，那么如何互相验证对方的证书?
  - CA中心应该能够互相验证对方的公钥

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.



包含两个CA中心 的CA 网络

- Alice:
  - 发送$CA_1(K_A^u)$ 和 $CA_2(K^u_{CA1})$给Bob
- Bob:
  - 用CA_2 的公钥验证CA_1的公钥
  - 用CA_1的公钥验证Alice的公钥

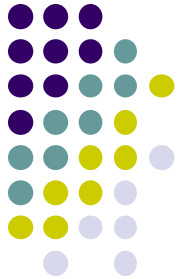


包含多个CA中心的CA 网络

- 从Alice 到 Bob: $CA_1 \rightarrow CA_5 \rightarrow CA_4$ and $CA_1 \rightarrow CA_3 \rightarrow CA_5 \rightarrow CA_4$
- 从Bob 到 Alice: $CA_4 \rightarrow CA_2 \rightarrow CA_1$

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.

钥匙环



- 系统可能有很多不同的用户
- 如何存储和管理这些公钥和私钥?
- 私钥环
  - 一个表，每一行是一个用户的信息：密钥ID，用户名，公钥，加密的私钥，时间戳等…
- 公钥环
  - 一个表，每一行是一个用户的信息：密钥ID，用户名，公钥，CA名称等…

《计算机网络安全的理论与实践（第2版）》.【美】王杰，高等教育出版社，2011年.