

公开（非对称）密码

范明钰

信息安全研究中心

要点

传统加密与公钥加密：动机和需求

应用举例；组成特点

RSA算法

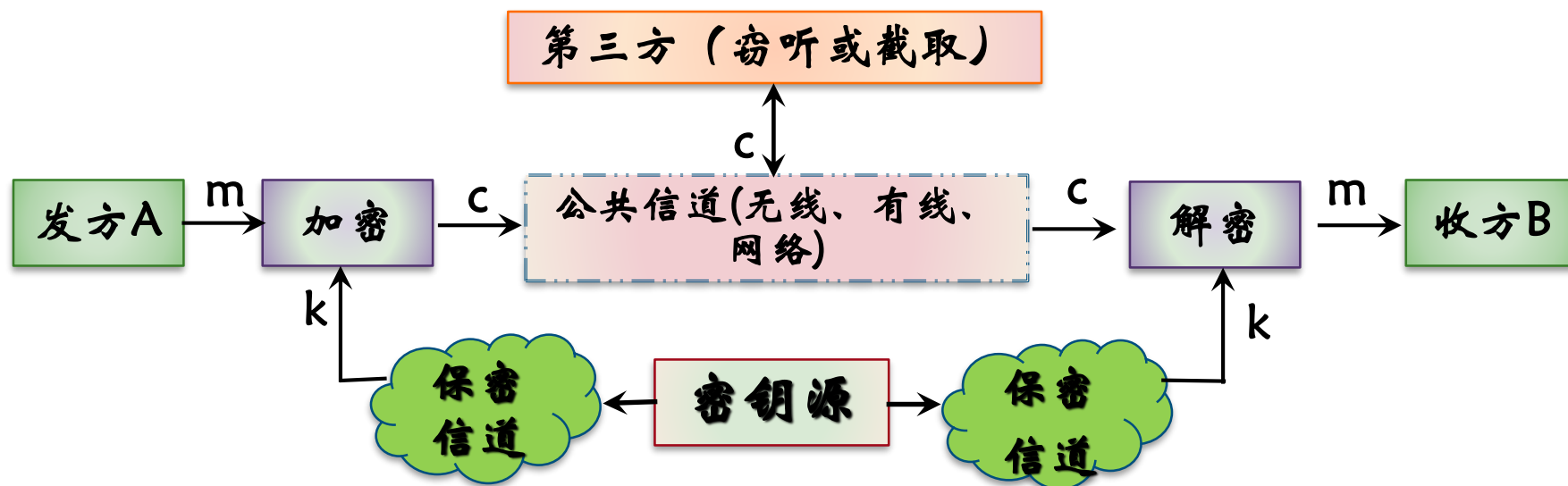
Elgamal算法

ECC算法

公钥应用：PKI

传统加密

单钥(传统)密码体制通信模型



问题1: 如何保证
信息 m 的**保密**?

问题2: 如何保证
信息 m 的**认证**?

问题3: 如何保证
信息 m 的**完整**?

动机和需求

◆ 对称密码算法的问题

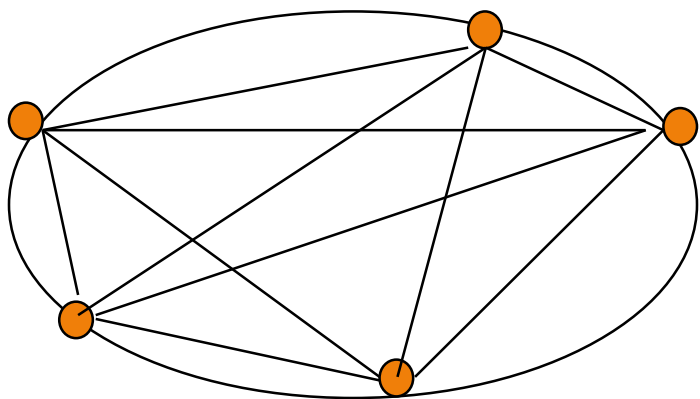
- 加密能力与解密能力捆绑
- 密钥管理问题：量、方法
- 其他安全需求：不相识、不信任的人之间通信的安全要求

对称密钥分配—量的需求

量的需求

密钥必须通过保密信道分配

密钥的数量 $O(n^2)$



对称密钥分配—解决方法

集中式密钥分配中心 (KDC)

每个用户和KDC之间共享一个主密钥,经可靠信道分配

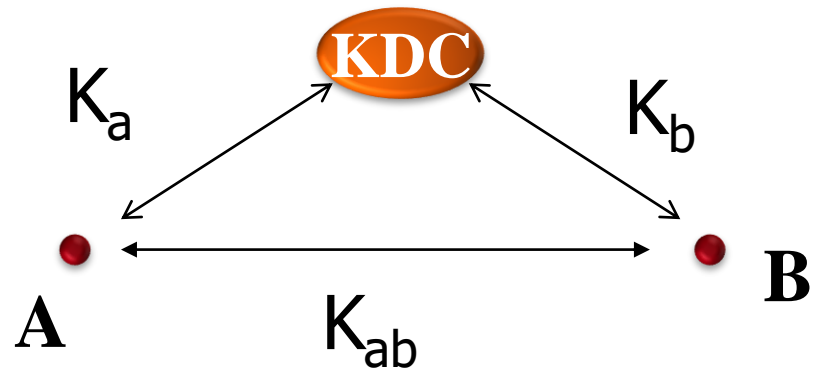
会话密钥协商

$A \rightarrow KDC$: 请求访问 B

$KDC \rightarrow A$: $K_a [K_{ab}], K_b [K_{ab}]$

$A \rightarrow B$: $K_b [K_{ab}]$

$A \leftrightarrow B$: $K_{ab}[m]$



公开密钥密码体制由来

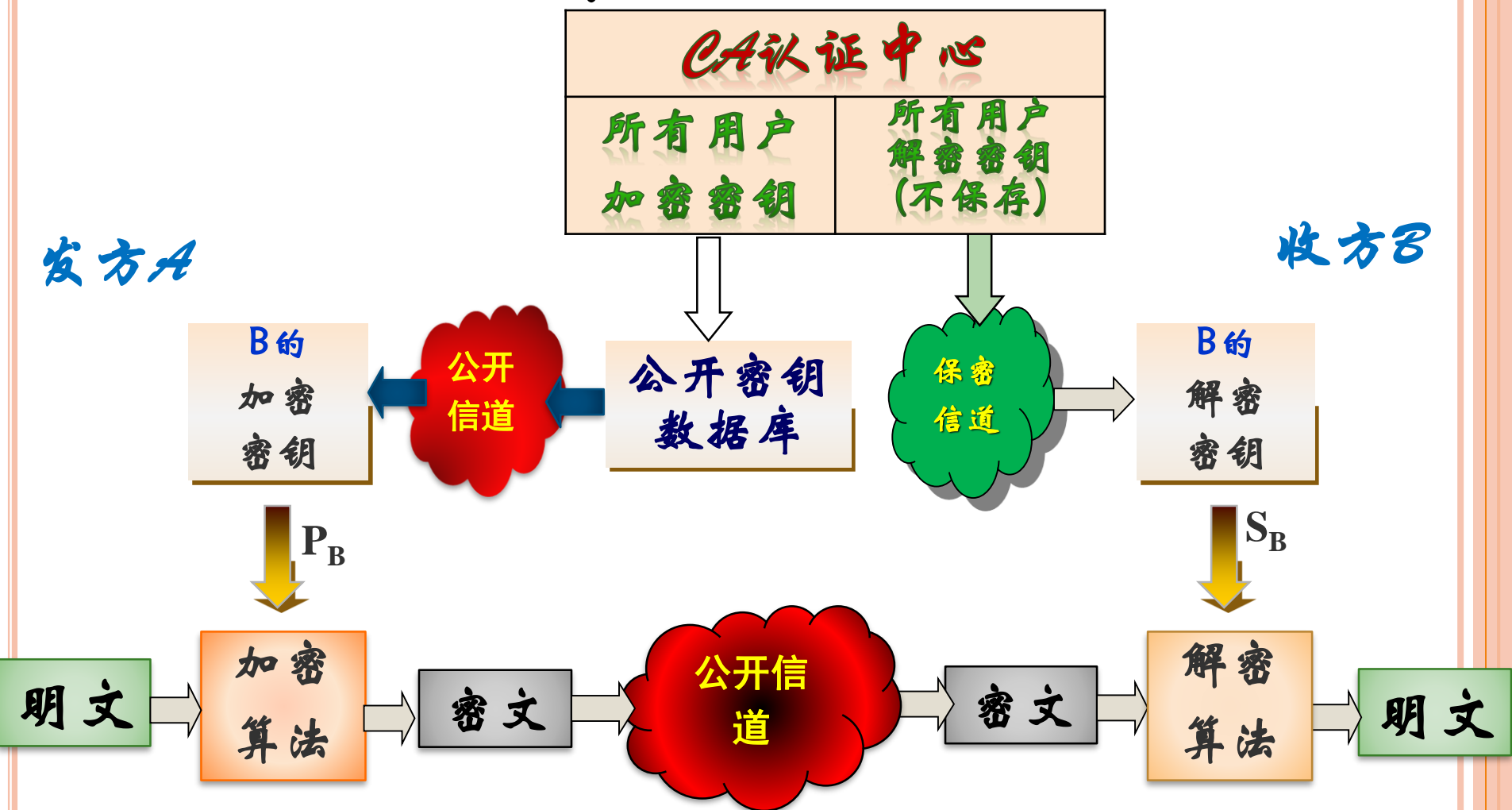
1976年, Diffie与Hellman发表了题为“New Direction in Cryptography”的文章, 提出:

构造这样的密码体制: 其加密算法和解密算法分别使用不同的密钥, 从而, 可以将加密密钥公开(称为公开密钥P)、解密密钥保密(称为保密密钥S)。要求:

公开密钥 P $\xrightarrow{\text{计算上不可行}}$ 保密密钥 S

根据上述密钥的特点, 形象地称传统密码体制为对称密钥密码体制, 公开密钥密码体制为非对称密钥密码体制。

公开密钥密码体制：应用



保密性体现: (对任何用户, 加密密钥 \longleftrightarrow 解密密钥)

计算上不可行

对公开钥密码算法—要求

◆ 计算容易

- 产生一对密钥(公钥 k_e 和私钥 k_d)在计算上是容易的
- 不难计算 $c=E_{k_e}(m)$ 和 $m=D_{k_d}(C)$

◆ 分析不可行

- 知道 k_e , 计算 k_d 不可行
- 不知道 k_d , 即使知道 k_e , E , D 及 C , 计算 m 不可行

◆ 加密变换和解密变换可以互换顺序

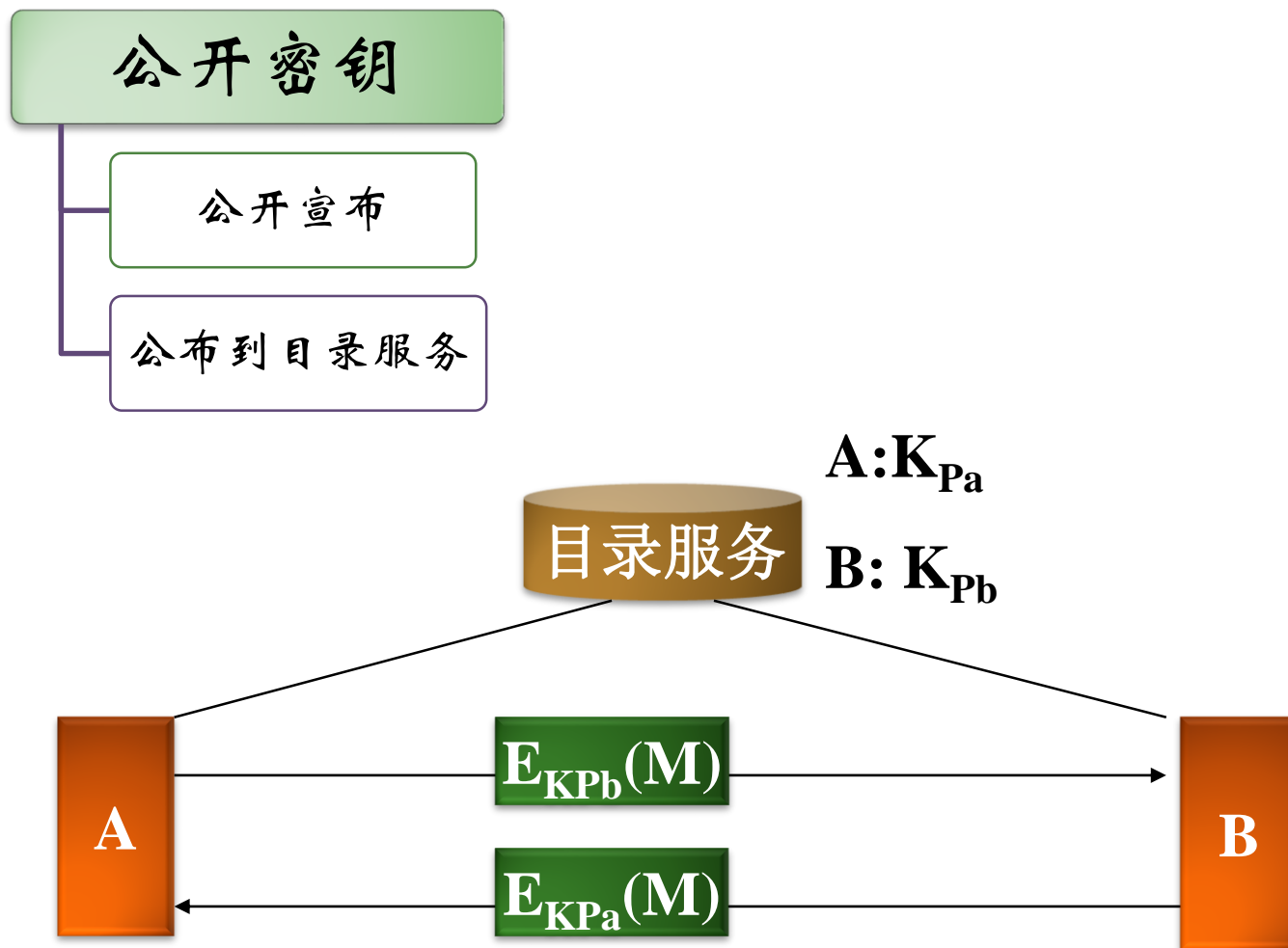
- 即 $D(E(m))=E(D(m))$

公开密钥密码体制—组成

◆ 组成

- 明文：算法的输入，可读信息或数据
- 加密算法：根据密钥，对明文进行转换，输出密文
- 公钥和私钥对：分别用于加密和解密
 - ✓ 每一用户产生一对密钥，用来加密和解密消息
 - ✓ 其中一个密钥（公钥）存于公开的寄存器或其他可访问的文件中；另一个密钥（私钥）秘密保存
- 密文：算法的输出，依赖于明文和密钥
- 解密算法：根据密钥，对密文进行处理，还原明文

公开密钥的密钥分配—优势



非对称密码体制的基本特点

- ◆ 加密与解密的能力是分开的
 - 通过密码算法和加密密钥来确定解密密钥，在计算上是不可行的
 - 两个密钥的任何一个都可用来加密，另一个用来解密。如何使用，取决于需求
- ◆ 密钥分发简单
- ◆ 需要保存的密钥量大大减少， N 个用户只需要 N 个密钥
- ◆ 可满足不相识的人之间保密通信
- ◆ 可以实现数字签名

对称和公开密钥加密的主要区别

◆ 对称密码

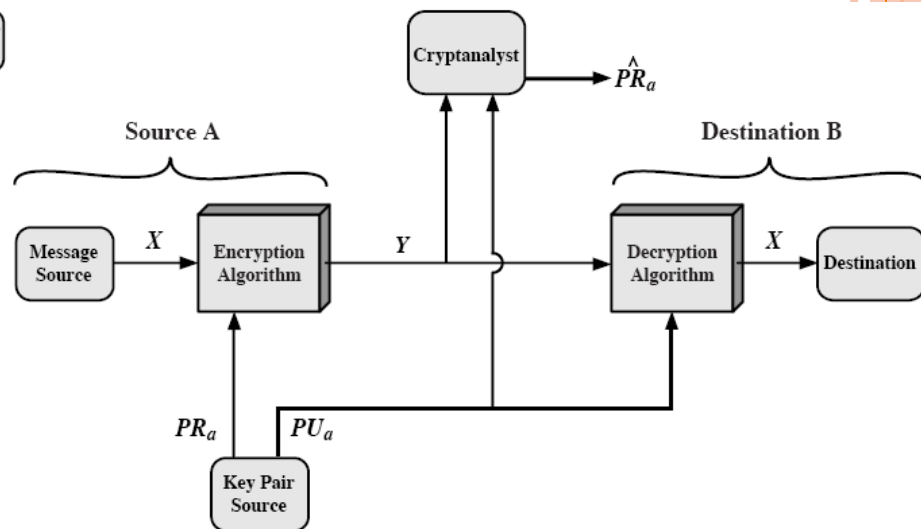
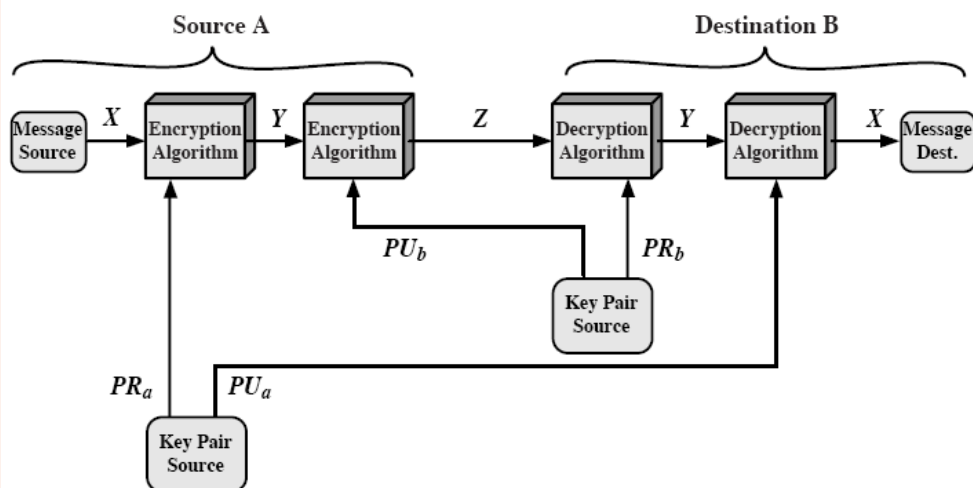
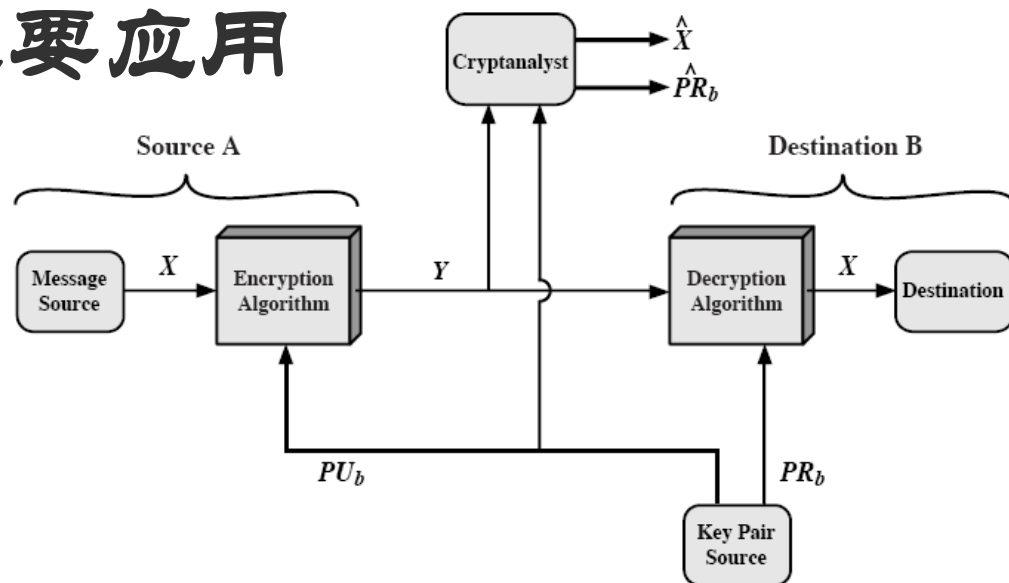
- 加/解密密钥相同，密钥为双方共享
- 密钥必须保密
- 已知算法和密文不易推导出密钥

◆ 公钥密码

- 加/解密密钥不同，双方各持一个。
- 其中一个密钥必须保密——私钥
- 已知算法和公钥，不易推导出私钥

公钥密码体制的主要应用

- ◆ 加密
- ◆ 认证
- ◆ 既加密又认证
- ◆ 密钥交换



公钥密码的常规分析

◆ 穷举密钥攻击，第一种

- 解决方法是使用长密钥
- 但是为了便于实现，又希望密钥足够短
- 目前仅限于密钥管理和签名

◆ 穷举消息攻击，是第二种攻击形式

- 攻击者用公钥对所有可能的消息加密，并与传送的密文匹配，从而解密任何消息
- 抵抗的方法是在要发送的消息后附加随机数。

◆ 从给定的公钥推算出私钥，是第三种攻击方法

- 多数公钥算法尚未在数学上证明可以抵抗这种攻击

依据的数学难题

1. 整数分解难题

RSA体制

背包问题：Merkle-Hellman体制

2 离散对数问题

2.1. 有限域的乘法群上的离散对数问题：
ElGamal公钥系统、Massey-Omura公钥系统

2.2. 椭圆曲线上的离散对数问题 (ECC)

发展情况

1978年以来，相继有：

- 基于大整数分解问题的RSA体制
- 基于背包问题的Merkle-Hellman体制
- 基于编码理论的McEliece体制
- 1985年，著名的基于离散对数问题的ElGamal体制
- 1986年，今天应用广泛的基于椭圆曲线上离散对数问题的椭圆曲线公开密钥密码体制

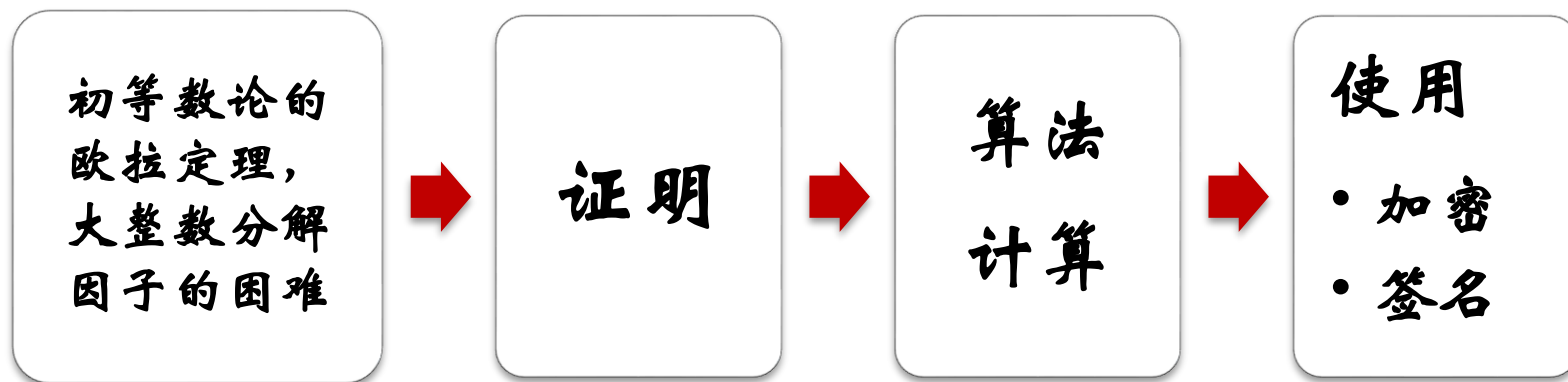
公钥密码算法代表

RSA 算法

Elgamal 算法

椭圆曲线算法 ECC

RSA算法



RSA公钥算法的数学基础--欧拉定理

表述1: 将 Z/n 表示为 Z_n , 其中
 $n=pq$; p, q 为素数且相异, 若 Z_n^*
 $\equiv \{g \in Z_n / (g, n) = 1\}$, 易见 Z_n^* 为 $\phi(n)$
阶的乘法群且有 $g^{\phi(n)} \equiv 1 \pmod{n}$,
而 $\phi(n) = (p-1)(q-1)$

表述2: 若整数 g 和 n 互素则 $g^{\phi(n)} \equiv 1 \pmod{n}$, 其中 $\phi(n)$ 为比 n 小
但与 n 互素的正整数个数, 称为 n 的
欧拉函数

RSA公开密钥密码体制—证明（略）

命题. 设 $0 \leq a < n$, 若 $r = k \pmod{\lambda}$, 则 $a^k \equiv a^r \pmod{n}$ 。

证明. $r = k \pmod{\lambda} \Leftrightarrow \lambda \mid (k-r) \Rightarrow (p-1) \mid (k-r)$ 。

设 $k = s(p-1) + r (s \geq 0)$, 有

$$a^k = a^{s(p-1)+r} = (a^{p-1})^s \cdot a^r$$

当 $a \neq 0$ 时, $(a, p) = 1$ 或 $(a, p) = p$,

根据上式和Fermat小定理显然有

$$a^k \equiv a^r \pmod{p}。$$

同理, $a^k \equiv a^r \pmod{q}。$

因为 $p \neq q$, 故 $a^k \equiv a^r \pmod{n}。$

RSA密码算法描述

前提：明文空间 $P = \text{密文空间 } C = \mathbb{Z}_n$ ，整数

1. 准备密钥：选择互异素数 p, q ，计算： $n = p * q$, $\phi(n) = (p-1)(q-1)$;
选择整数 e 使 $(\phi(n), e) = 1, 1 < e < \phi(n)$), 计算 d , 使 $d = e^{-1} \pmod{\phi(n)}$ 。
得公钥 $P_k = \{e, n\}$; 私钥 $S_k = \{d, n\}$

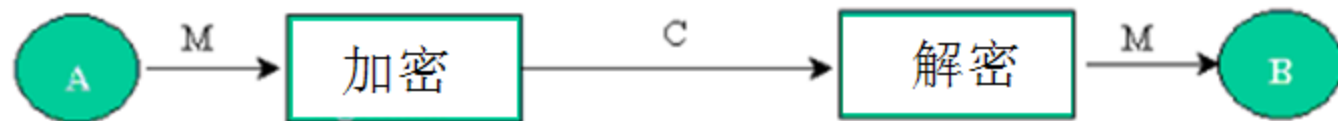
2. 加密(用 e, n): 明文 $M < n$ ，密文 $C = M^e \pmod{n}$

3. 解密(用 d, n): 密文 C ，明文 $M = C^d \pmod{n}$

例子 1

RSA (Rivest - Shamir - Adleman)

Public key encryption algorithm (Asymmetric), 1977



Algorithm:

→ key generation

Step1: Choose p, q where p and q are prime, and calculate $n=pq$.

Step2: Select e such that $\gcd(e, \phi(n))=1$
where $\phi(n)=(p-1)(q-1)$

Step3: Calculate $d=e^{-1} \bmod n$

Step4: your public key $\{e, n\}$: public

your secret key $\{d, n\}$: keep secret by yourself

→ Encryption

$$C = M^e \bmod n$$

→ Decryption

$$M = C^d \bmod n$$

where M : Plaintext and C : Ciphertext

→ key-length ≥ 512 bits

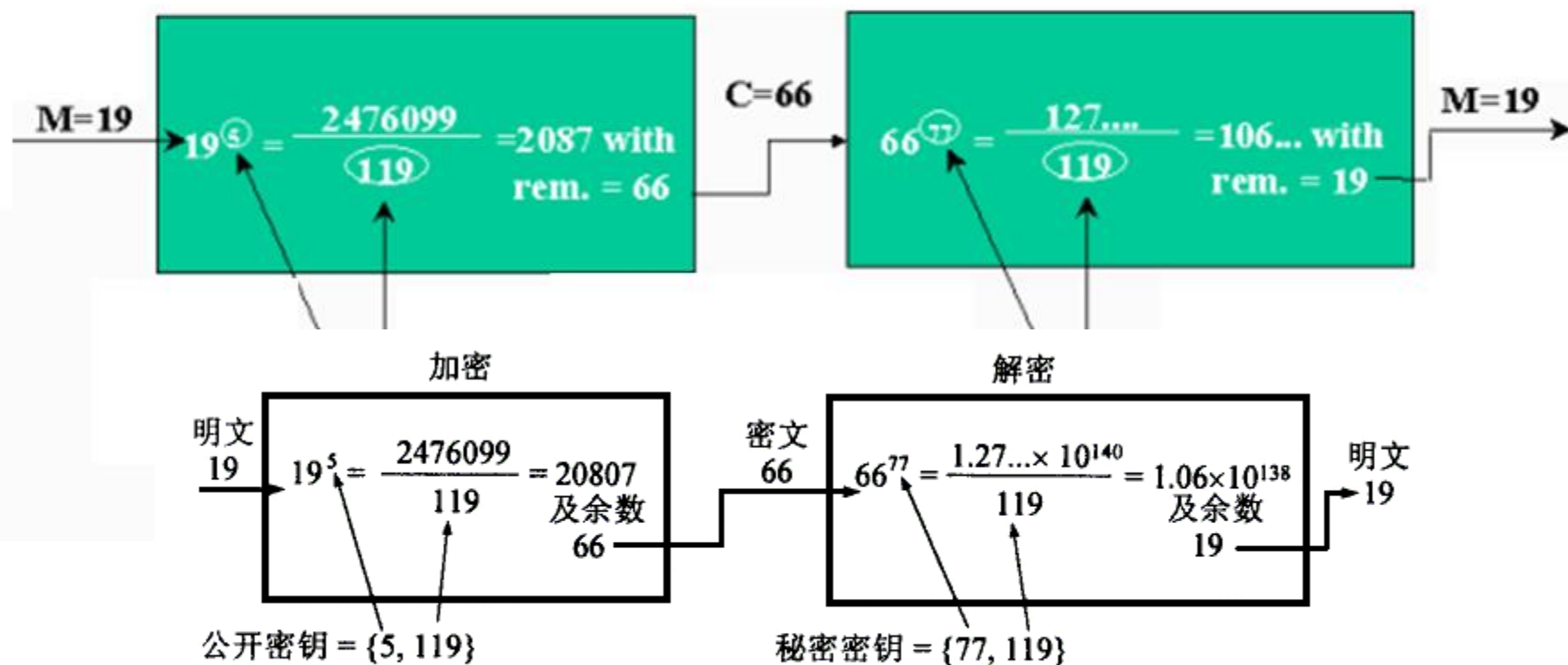
(B, e_B, n_B)



RSA (Rivest - Shamir - Adleman)

Example:

- 1: Choose $p=7$ and $q=17$.
- 2: Calculate $n=pq=7 \times 17=119$.
- 3: Calculate $\phi(n)=(p-1)(q-1)=96$.
- 4: Select $e=5$ (relatively prime to $\phi(n)$).
- 5: Determine d such that $de=1 \pmod{96}$ and $d < 96$; $d=77$. since $77 \times 5 = 4 \times 96 + 1$.



RSA公开密钥密码体制—计算

可用上述命题简化计算 $a^k(\bmod n)$ 。

例：设 $p=13, q=17$ ，那么 $n=13 \times 17=221$ 。试对于 $a=11$ ，计算 $a^{51}(\bmod 221)$ 。

解：因为 $\varphi=[p-1, q-1]=[12, 16]=48$ ，

$51 = 3(\bmod 48)$ ，则 $a^{51} = a^3 = 1331 = 5(\bmod 221)$ 。

■ 计算模 n 的幂

RSA体制的加/解密算法都涉及到：

对正整数 k 及整数 a ，计算 $a^k(\bmod n)$

由前面的命题，根据 $n=pq$ (p, q 为素数)的特点，可将指数 k 减小至 $\varphi=[p-1, q-1]$ 以下；但接下来的计算还须讲究方式

RSA公开密钥密码体制—计算

- 计算 $a^k \pmod n$ 的自然方法：

Input: k, a, n

Set $x=1$

For $i=1, 2, \dots, k$ repeat


Set $x=xa \pmod n$

Output: x

RSA公开密钥密码体制—计算

◆ 计算 $a^k \pmod n$ 的快速算法：设 $k=(1k_{t-1}\cdots k_1k_0)_2$

——从左向右计算方法(“平方-条件乘”)：

$$a^k = (\cdots((a^2 \cdot a^{k_{t-1}})^2 \cdot a^{k_{t-2}})^2 \cdots a^{k_1})^2 \cdot a^{k_0}$$


Input: $t, k_0, k_1, \cdots, k_{t-1}, a, n$

Set $x=a$

For $i=t-1, t-2, \cdots, 1, 0$ repeat

Set $x=x^2 \pmod n$

If $k_i=1$ then set $x=xa \pmod n$

Output: x

RSA公开密钥密码体制—计算

——从右向左计算方法(“条件乘-平方”) 设 $k=(1k_{t-1}\dots k_1k_0)_2$

$$\leftarrow a^k = a^{2^t} \cdot (a^{2^{t-1}})^{k_{t-1}} \cdot \dots \cdot (a^{2^2})^{k_2} \cdot (a^{2^1})^{k_1} \cdot (a^{2^0})^{k_0} \cdot 1$$

Input: $t, k_0, k_1, \dots, k_{t-1}, a, n$

Set $y=a, x=1$

For $i=0, 1, \dots, t-1$ repeat

 If $k_i=1$ then set $x=yx \pmod n$

 Set $y=y^2 \pmod n$

Set $x=yx \pmod n$

Output: x

RSA公开密钥密码体制—计算

■ 举例

① $p=7, q=13$; $n=\underline{91}, \Phi(n)=\underline{72}$; 若取

$d=31$, 则 $e=\underline{7}$;

公开密钥 $P_B = \underline{(7, 91)}$,

加密算法: $\underline{c=m^7(\text{mod}91)}$;

保密密钥 $S_B = \underline{31}$,

解密算法: $\underline{m=c^{31}(\text{mod}91)}$ 。

例如, $m=11$, 可以算出 $c=\underline{67}$ 。

RSA公开密钥密码体制—计算

② $p=47$, $q=59$; $n=\underline{2773}$, $\Phi(n)=\underline{2668}$; 若

取 $e=157$, 则 $d=\underline{17}$;

公开密钥 $P_B = \underline{(157, 2773)}$,

加密算法: $\underline{c=m^{157}(\text{mod}2773)}$;

保密密钥 $S_B = \underline{17}$,

解密算法: $\underline{m=c^{17}(\text{mod}2773)}$ 。

例如, $c=192$, 可以算出 $m=\underline{920}$ 。

RSA使用问题

选择素数

检测素数

计算密钥

1. Select p, q p and q both prime
2. Calculate $n = p \times q$
3. Calculate
4. Select integer e
5. Calculate d
6. Public Key $KU = \{e, n\}$
7. Private key $KR = \{d, n\}$

生成素数对

① 随机选取大约用 $\ln(N)/2$ 的次数，如 $\ln(2^{200})/2=70$

② 在生成密钥对时才用到，慢一点可忍受

③ 确定素数 p 和 q 以后，只需选取 e ，满足 $\gcd(e, \phi(n))=1$ ，计算 $d = e^{-1} \bmod \phi(n)$ （sieve 扩展的欧拉算法）

选择素数

目前还没有高效的办法。实际中应用最多的是Miller and Rabin, WITNESS算法



通常，随机选取一个大的奇数，然后进行素性检验

素数检测理论和方法

直接判断一个整数是
否为素数是困难的

命题: 如果 p 是素数,
则方程 $x^2 \equiv 1 \pmod{p}$
只有平凡解 $x \equiv \pm 1$
 \pmod{p} .

• 证明: 略

若方程 $x^2 \equiv 1 \pmod{p}$
的解不是 $x \equiv \pm 1$
 \pmod{p} , 则 p 不是素
数

WITNESS 算法

- ◆ 判断给定数字 n 是否是素数，比较简单的方法是试除，试着用 $2, 3, \dots$ 直到根号 n 去除 n 。显然，如果 n 是素数，则没有一个数能整除 n ；减少不少除法的次数的改进是，只用2到根号 n 的素数做为除数，但效率不高
- ◆ Witness算法：在不找出因子的情况下，判断 n 是否为素数重复 n 次实验。对于每一次实验，随机取检验算子 a ，看看在算子 a 下， n 能否满足：
 - $a^r \equiv 1 \pmod n$ 或者对某个 j ($0 \leq j \leq s-1, j \in \mathbb{Z}$) 等式 $a^{(2^j)r} \equiv -1 \pmod n$
- ◆ 如果任意一次实验不满足，则判定不是素数，如果都满足，可近似可以认为是素数（错误率极小）

WITNESS(A,N)算法

```
1. 令 $b_k b_{k-1} \dots b_0$  为 $(n-1)$ 的二进制表示,  
2.  $d \leftarrow 1$   
3. for  $i \leftarrow k$  downto 0  
4.   do  $x \leftarrow d$   
5.    $d \leftarrow (d \times d) \bmod n$   
6.   if  $d = 1$  and  $x \neq 1$  and  $x \neq n-1$   
7.     then return TRUE  
8.   if  $b_i = 1$   
9.     then  $d \leftarrow (d \times a) \bmod n$   
10. if  $d \neq 1$   
11. then return TRUE  
12. return FALSE
```

判定 n 是否为素数

设 a 是小于 n 的整数

返回值: TRUE
则 n 一定不是素数, FALSE则 n 可能是素数

应用

随机选择 $a < n$, 计算 s 次,

如果每次都返回 FALSE, 则这时 n 是素数的概率为 $(1 - 1/2^s)$

RSA算法的**应用**--密钥交换

- ◆ Diffie-Hellman 密钥交换：是第一个公钥方案，
Diffie & Hellman in 1976, now know that James Ellis
(UK CESC) secretly proposed the concept in 1970
- ◆ 密钥交换方案
 - 允许两个用户安全地建立秘密信息，用于后续的通讯过程
 - 该秘密信息仅为两个参与者知道
- ◆ 在美国的专利1997年4月29日到期

*DIFFIE-HELLMAN*密钥交换算法

◆ 双方准备:

- AB双方选择: 素数 p 以及 p 的一个原根 a (也可由一方选择后发给对方)
- 用户A: 选择一个随机数 $X_a < p$, 计算 $Y_a = a^{X_a} \bmod p$
- 用户B: 选择一个随机数 $X_b < p$, 计算 $Y_b = a^{X_b} \bmod p$
- 双方各保密自己的 X 值

◆ 传送: 双方将计算出的 Y 值给对方

◆ 双方计算:

- 用户A计算出 $K = Y_b^{X_a} \bmod p$
- 用户B计算出 $K = Y_a^{X_b} \bmod p$

◆ 这样, AB双方获得一个共享密钥($a^{X_a X_b} \bmod p$)

对DIFFIE-HELLMAN密钥交换的攻击

◆ 中间人攻击

- ① 双方选择素数 p 以及 p 的一个原根 a (假定 O 知道)
- ② A 选择 $X_a < p$, 计算 $Y_a = a^{X_a} \bmod p$, $A \rightarrow B: Y_a$
- ③ O 截获 Y_a , 选 X_o , 计算 $Y_o = a^{X_o} \bmod p$, 冒充 $A \rightarrow B: Y_o$
- ④ B 选择 $X_b < p$, 计算 $Y_b = a^{X_b} \bmod p$, $B \rightarrow A: Y_b$
- ⑤ O 截获 Y_b , 冒充 $B \rightarrow A: Y_o$
- ⑥ A 计算: $(Y_o)^{X_a} \equiv (a^{X_o})^{X_a} \equiv a^{X_o X_a} \bmod p$
- ⑦ B 计算: $(Y_o)^{X_b} \equiv (a^{X_o})^{X_b} \equiv a^{X_o X_b} \bmod p$
- ⑧ O 计算: $(Y_a)^{X_o} \equiv a^{X_a X_o} \bmod p$, $(Y_b)^{X_o} \equiv a^{X_b X_o} \bmod p$

◆ 攻击成功的条件:

- ① O 必须实时截获并冒充转发, 否则会被发现

安全级别比较

- ◆ 公开密钥算法的密钥长度随着保密要求提高，增加很快

保密级别	对称密钥长度 (bit)	RSA密钥长度 (bit)	ECC密钥长度 (bit)	保密年限
80	80	1024	160	2010
112	112	2048	224	2030
128	128	3072	256	2040
192	192	7680	384	2080
256	256	15360	512	2120

对公钥密码算法的误解：

公钥算法VS 单钥算法

更安全？

任何算法都依赖于
密钥长度、破译的
工作量，从抗分析
角度，没有一方更
优越

单钥算法过时
了？

公钥算法很慢，只
能用在密钥管理和
数字签名
单钥算法将长期存
在

密钥分配更简
单？

事实上，既
不简单，也
不有效

对RSA的分析

1、对n作大质因子分解：建立质数数据库

2、选择密文攻击：
提供伪装信息诱使对方以私钥加密后传回
 $(XM)^d = X^d * M^d \% n$ 。解决方法：不以私钥对来自外面的信息作加密

3、共享n攻击： $C_1 = M^{e_1} \% n$ ， $C_2 = M^{e_2} \% n$ ； $r * e_1 + s * e_2 = 1$ (质数性质)；
 $m = (C_1^{-1})^{-r} * (C_2^{-1})^{-s} \bmod n$

上世纪90年代大数分解的进程

- ◆ 分解数 尺寸bits 分解日期 分解算法
- ◆ RSA-100 330 1991.4 二次筛法 ← 1981年, <100位 十进制数
- ◆ RSA-110 364 1992.4 二次筛法
- ◆ RSA-120 397 1993.6 二次筛法
- ◆ RSA-129 425 1994.4 二次筛法
- ◆ RSA-130 430 1996.4 数域筛法 ← 1993年, >110位
- ◆ RSA-140 463 1999.2 数域筛法
- ◆ RSA-155 512 1999.8 数域筛法

RSA: 问题

生成密钥很麻烦，难以做到每次都选取不同的 p 、 q

计算能力提升要求 n 必须越来越大，导致 p 、 q 也跟着变大

最常见的解决之道是RSA + DES + Hash合并应用

作业

1. 对用户A的RSA密码体制构造如下：

$p=11$, $q=13$; $n=$ _____, $\Phi(n)=$ _____；若取 $e=17$, 则
 $d=$ _____；

公开密钥 $P_A=$ _____，

加密算法：_____；

保密密钥 $S_A=$ _____，

解密算法：_____。

作业

2. 对用户B的RSA密码体制构造如下：

$p=13, q=17$; $n=$ _____, $\Phi(n)=$ _____; 若取 $e=13$, 则 $d=$ _____;

公开密钥 $P_B=$ _____,

加密算法: _____;

保密密钥 $S_B=$ _____,

解密算法: _____。

3. 在RSA公开密钥密码体制中，用户A和用户B的密钥如上面1题和2题，若用户A要将 $m=23$ 加密后发给用户B，求相应的密文 $c=$ _____。

答案

1. $n=143, \Phi(n)=120,$

$$\begin{aligned} 120 &= 7 \times 17 + 1 \\ 17 &= 17 \times 1 \end{aligned} \quad \begin{pmatrix} 0 & 1 \\ 1 & -7 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -17 \end{pmatrix} = \begin{pmatrix} * & * \\ -7 & * \end{pmatrix}$$

$$d = -7 = 120 - 7 = 113 \pmod{120}$$

公开密钥 $P_A = (17, 143)$, 保密密钥 $S_A = 113$

2. $n=221, \Phi(n)=192,$

因为 $192 = 14 \times 13 + 10$

$$13 = 1 \times 10 + 3$$

$$10 = 3 \times 3 + 1$$

$$3 = 3 \times 1$$

所以 $d = -59 = 192 - 59 = 133$

公开密钥 $P_A = (13, 221)$, 保密密钥 $S_A = 133$

3. $C = 23^{13} = ((23^2)^2)^2 ((23^2)^2 23 = 152 \times 55 \times 23 = 183 \times 23 = 10 \pmod{221})$

$$(23^2 = 87, ((23^2)^2 = 55, ((23^2)^2)^2 = 152)$$

应用标准介绍：公钥证书（CERTIFICATE）

○ 公钥证书的主要内容

○ 身份证主要内容

持有者（Subject）标识	→	姓名
序列号	→	身份证号码
公钥（n,e）	→	照片
有效期	→	有效期
签发者（Issuer）标识	→	签发单位
CA的数字签名	→	签发单位盖章、防伪标志

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=CN, ST=Beijing, L=Tsinghua, O=CERNET, OU=CCERT,
CN=ccert-ca/Email=dhx@ccert.edu.cn

Validity

Not Before: Aug 27 22:17:42 2002 GMT

Not After : Aug 27 22:17:42 2003 GMT

Subject: C=CN, ST=Beijing, L=Tsinghua, O=CERNET, OU=ccert,
CN=ra.ccert.edu.cn/Email=dhx@cernet.edu.cn

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:c6:98:53:55:cb:61:d3:50:18:29:6d:37:55:e2:
36:d2:90:5f:f0:d4:cb:28:74:a7:38:b0:b3:5e:84:
71:97:76:65:d3:3c:59:8a:35:a8:85:86:64:5c:fc:
78:3d:c4:d0:23:52:1b:f9:6c:35:0e:b8:73:ad:ac:
33:1d:5d:dd:88:11:60:eb:24:88:30:38:12:63:50:
96:cb:8e:84:4b:f1:c8:a0:60:1c:e1:b3:b9:c0:cf:
19:20:50:8d:8a:97:eb:91:94:56:06:12:bd:9a:2c:
1a:b4:11:23:cd:a4:f8:a3:ce:de:f8:47:6b:0a:fe:
13:9b:89:e7:f1:96:e7:60:9d

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Cert Type:

SSL Client, S/MIME, Object Signing

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

EE:73:AA:36:92:D3:19:78:2F:5E:4A:04:38:7C:57:95:8D:14:0A:E1

X509v3 Authority Key Identifier:

keyid:89:87:06:B7:E1:95:7B:19:64:92:5A:9B:7C:22:71:27:7B:97:5E:A8

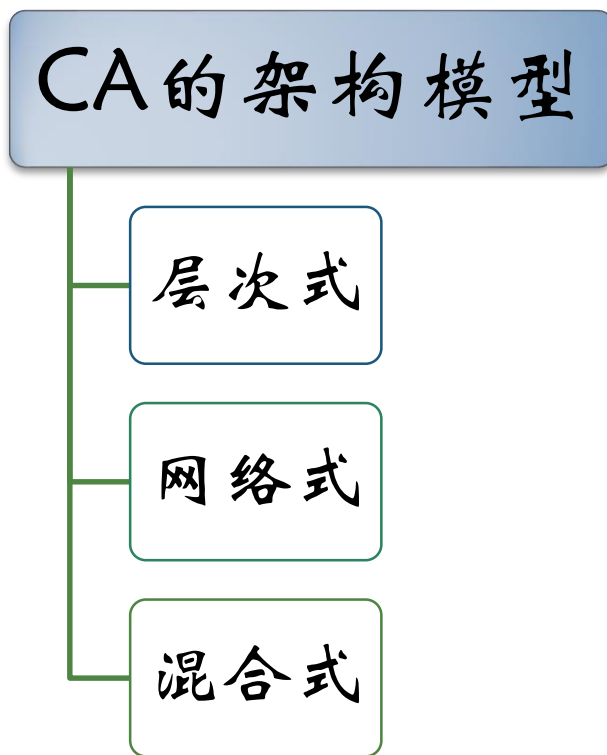
DirName:/C=CN/ST=Beijing/L=Tsinghua/O=CERNET/OU=CCERT/
CN=ccert-ca/Email=dhx@ccert.edu.cn

serial:00

Signature Algorithm: md5WithRSAEncryption

3a:53:95:a0:e6:d5:dd:17:16:c5:61:22:f8:15:69:2f:b9:b6:
25:e8:69:36:c5:cf:f2:f0:67:c8:5d:76:ad:1e:2c:29:b0:9e:
86:38:fa:69:89:62:6b:f5:71:ff:52:ee:18:1b:36:ed:a0:3f:
81:19:6e:e6:86:fc:c3:b5:5a:e5:df:38:6c:25:21:90:d1:80:
f2:fc:6a:4a:a5:83:9d:50:8b:d2:72:32:c1:4b:61:5e:32:38:
2e:73:46:ac:fb:38:6d:a7:ff:75:38:6a:c8:8f:89:90:81:31:
e6:e9:75:8f:2d:8b:0a:bb:fb:c6:b2:b5:1d:54:96:08:b8:98:
e1:23

证书中心架构分类



PKI/CA标准与协议

基础标准/协议

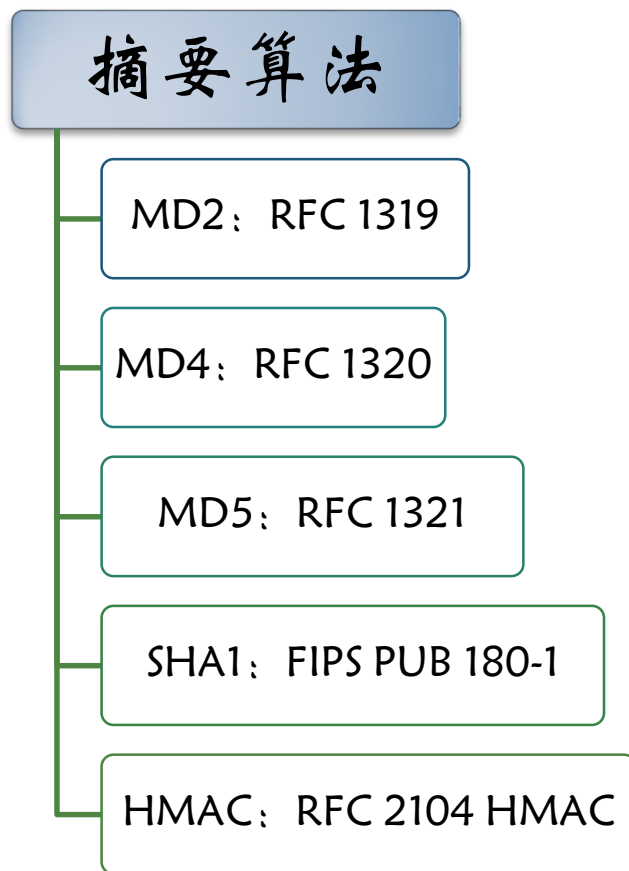
证书和CRL标准

操作标准/协议

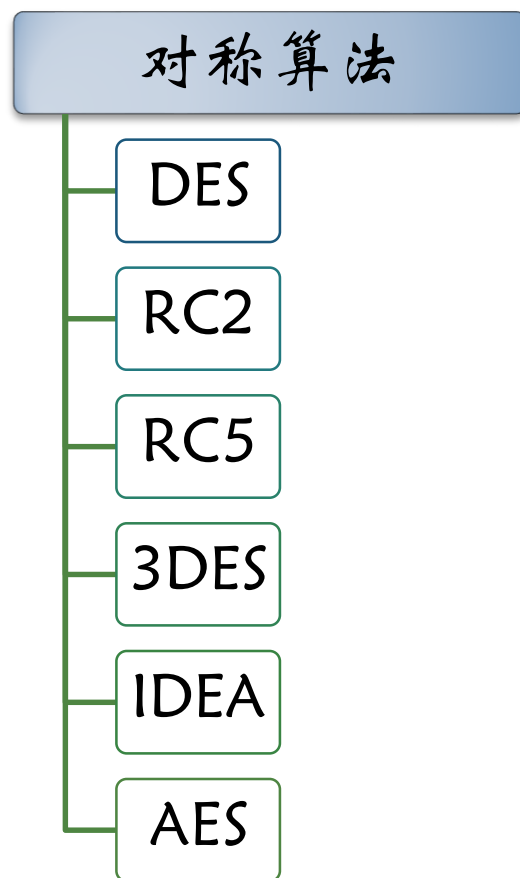
管理标准/协议

应用标准/协议

PKI/CA标准与协议：基础标准/协议



PKI/CA标准与协议



PKI/CA标准与协议

非对称算法

```
graph TD; A[非对称算法] --- B[RSA]; A --- C[DSA: 只用于签名]; A --- D[DH: 只用于密钥交换];
```

RSA

DSA: 只用于签名

DH: 只用于密钥交换

PKI/CA标准与协议

ASN.1 (Abstract Syntax Notation One) 抽象语法描述，是描述在网络上传输信息格式的标准方法

BER(基本编码规则)

CER(正规编码规则)

DER(可辨别编码规则)

PKI/CA标准与协议

<http://www.rsasecurity.com/rsalabs/pkcs>

PKCS系列标准

PKCS#1: RSA 加密标准

PKCS#3: Diffie-Hellman 密钥交换标准

PKCS#5: 基于口令的标准

PKCS#6: 基于证书的语义

PKCS#7: 消息语义

PKCS#8: 私钥语义

PKCS#9: 选择属性

PKCS#10: 凭证请求

PKCS#11: 加密标记接口

PKCS#12: 个人信息交换

PKCS#13: ECC标准

PKCS#14

PKCS#15

加密标准

PKI/CA标准与协议

GSS-API v2.0

GCS-API

CDSA

RSA PKCS#11 Cryptographic Token
Interface Standard v2.01

RSA BSAFE API

MS CryptoAPI v2.0

CTCA 证书存储介质接口规范v1.0

PKI/CA标准与协议

证书和CRL标准/协议

RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile: 描述 X.509 V3公钥证书和X.509 V2 CRL格式

RFC 3280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile: 对 RFC2459的更新, 增加或细化了证书路径验证算法、利用 CRL确定证书状态的算法、增量CRL、扩展项方面的描述

RFC2528 Internet X.509 Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates: 定义了一些对象标志, 并描述了密钥交换算法

PKI/CA标准与协议

操作标准/协议

LDAP: RFC 1777, Lightweight Directory Access Protocol

RFC 2587 Internet X.509 PKI LDAPv2

HTTP: 2616 Hypertext Transfer Protocol -- HTTP/1.1

FTP

OCSP: RFC 2560, X.509 Internet PKI Online Certificate Status Protocol

RFC 2559 Internet X.509 PKI Operational Protocols - LDAPv2

RFC 2585 Internet X.509 PKI Operational Protocols: FTP and HTTP

6

PKI/CA标准与协议

管理标准/协议

CRMF : FRC 2511, Internet X.509 Certificate Request Message Format

CMP: RFC 2510, Internet X.509 PKI Certificate Management Protocols

IKE: RFC 2409 The Internet Key Exchange

CP: RFC 2527, Internet X.509 PKI Certificate Policy and Certification Practices Framework

3029 Internet X.509 PKI Data Validation and Certification Server Protocols

3039 Internet X.509 PKI Qualified Certificates Profile

3161 Internet X.509 PKI Time-Stamp Protocol (TSP)

PKI/CA标准与协议

管理标准/协议

RFC 2528 Representation of Key Exchange Algorithm (KEA) keys in Internet X.509 Public Key Infrastructure Certificates

RFC 2538 Storing Certificates in the Domain Name System (DNS)

PKI/CA标准与协议

应用标准/协议

SSL/TLS: RFC 2246 The TLS Protocol Version 1.0

SET: Security Electronic Transaction

S/MIME: RFC 2312 Version 2 Certificate Handling

IPSec

PGP

WAP

PKI/CA标准与协议

应用标准：SSL/TLS

是Netscape公司设计用于Web安全传输协议

IETF将其标准化，成为RFC2246

分为两部分：握手协议和记录协议

握手协议负责协商密钥，协调客户和服务端使用的安全级别并进行身份认证

记录协议定义传输的格式和加解密应用程序协议的字节流

使用第三方证书机构提供的证书是SSL安全功能的基础

PKI/CA标准与协议

应用标准：SET

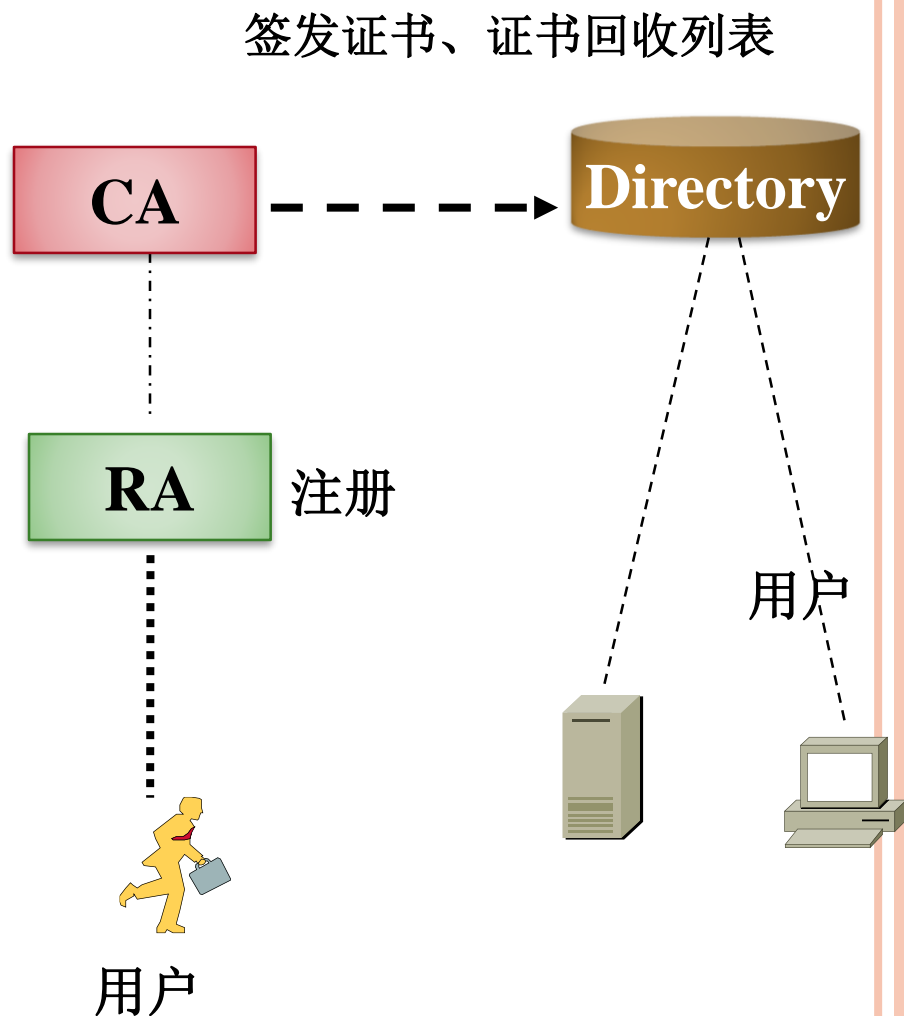
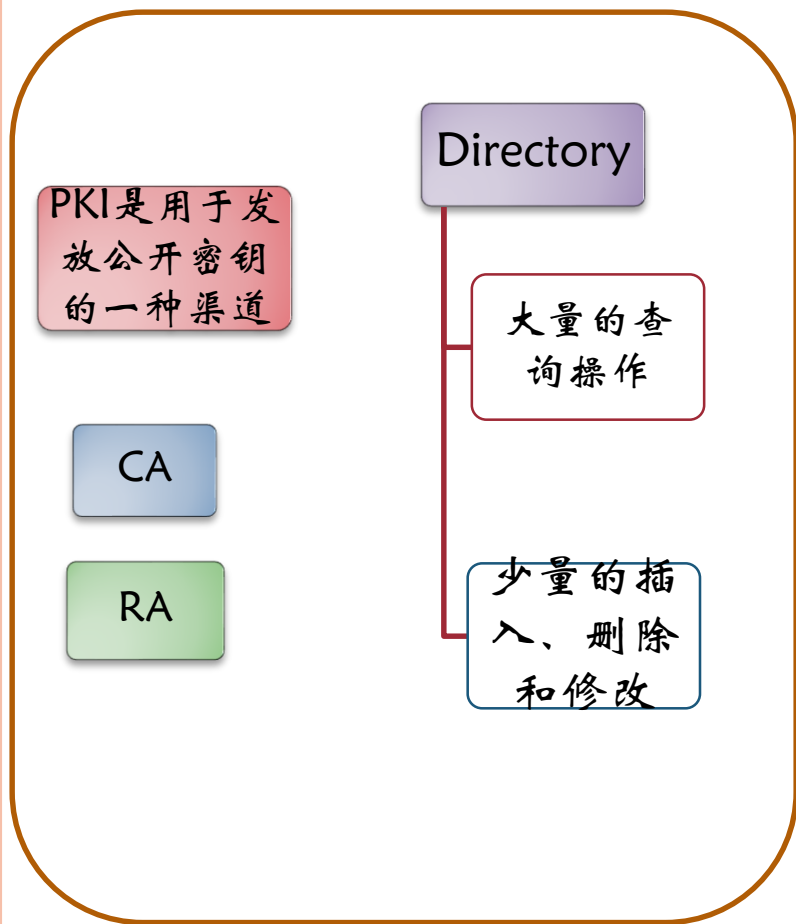
Visa, Master信用卡公司发起的在Internet网上实现安全电子商务交易系统的协议和标准

规定了交易各方使用证书方式进行安全交易的具体流程，及Internet上用于信用和支付的证书的发放和处理协议

主要技术：对称密钥加密、公共密钥加密、哈希算法、数字签名技术以及公共密钥授权机制等

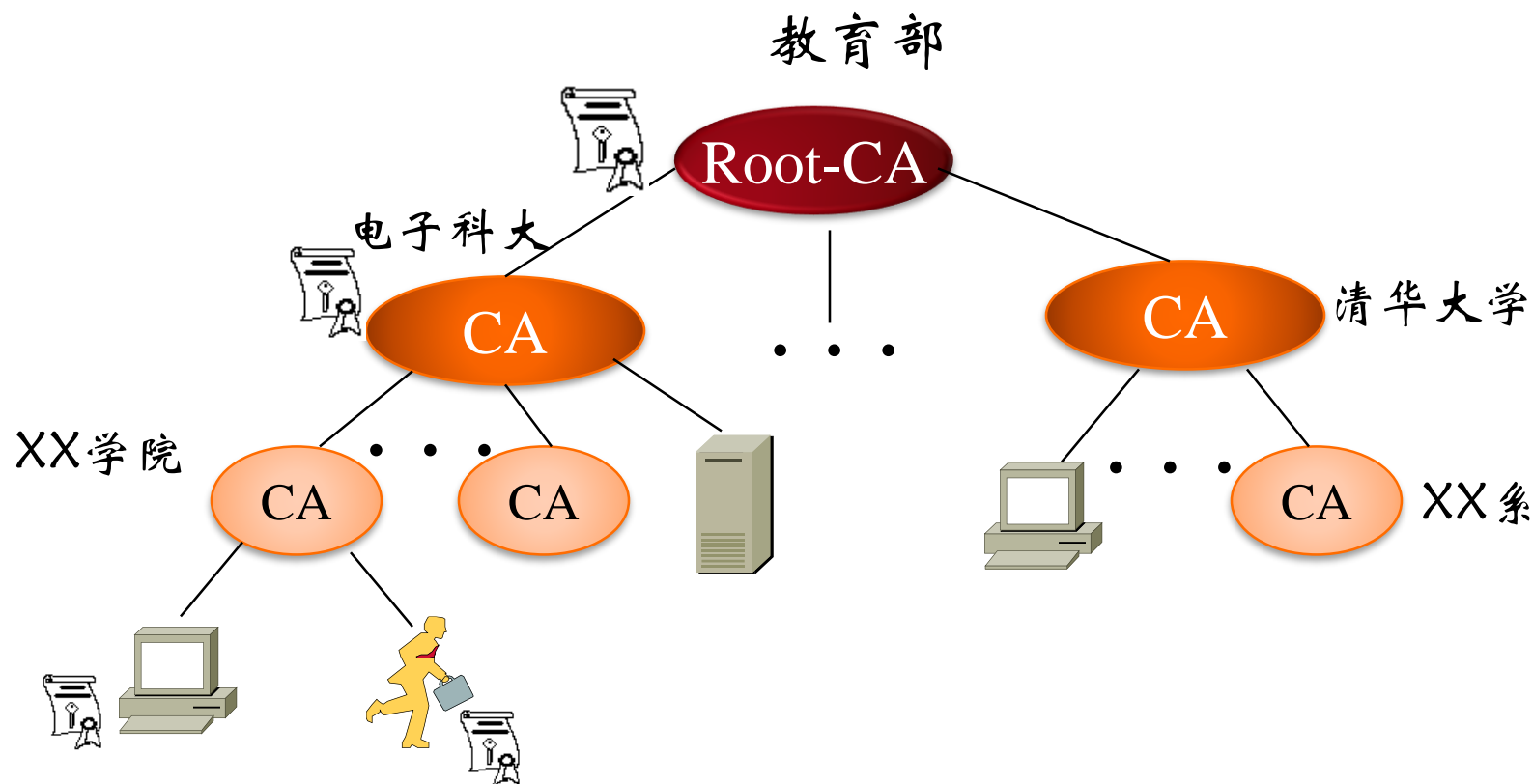
由两部分组成：证书管理和支付系统

公钥基础设施(PKI)



公钥基础设施 (PKI)

◆ X509 证书的层次管理结构：举例



下次内容

◆ Elgamal 算法