密 码 学

第九讲　公钥密码（1）

张焕国

武汉大学计算机学院

空天信息安全与可信计算教育部重点实验室

内容简介

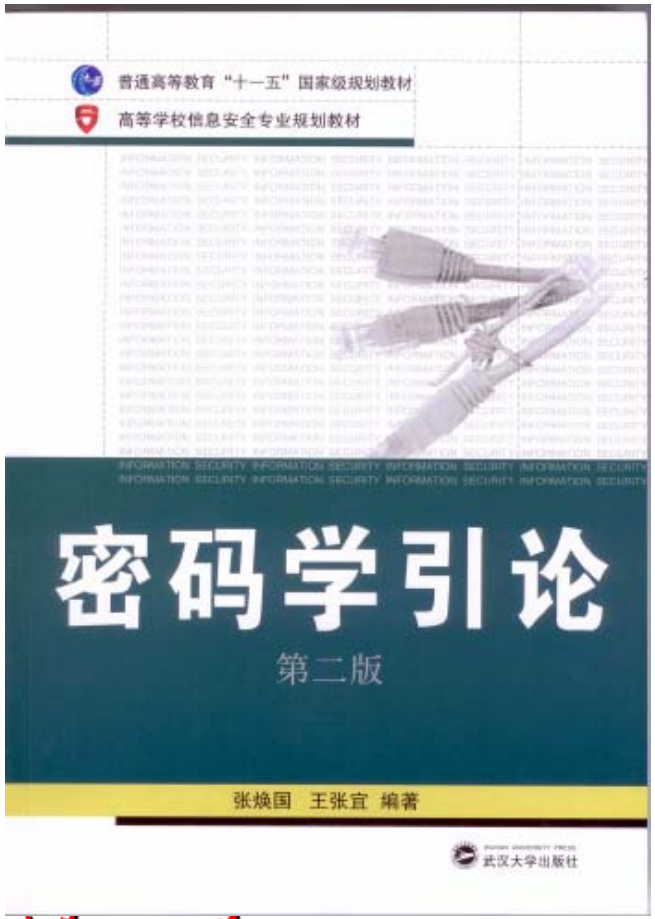
第一讲 信息安全概论
第二讲 密码学的基本概念
第三讲 数据加密标准（DES）
第四讲 高级数据加密标准（AES）
第五讲 中国商用密码（SMS4）
第六讲 分组密码的应用技术
第七讲 序列密码
第八讲 复习
第九讲 公钥密码（1）

内容简介

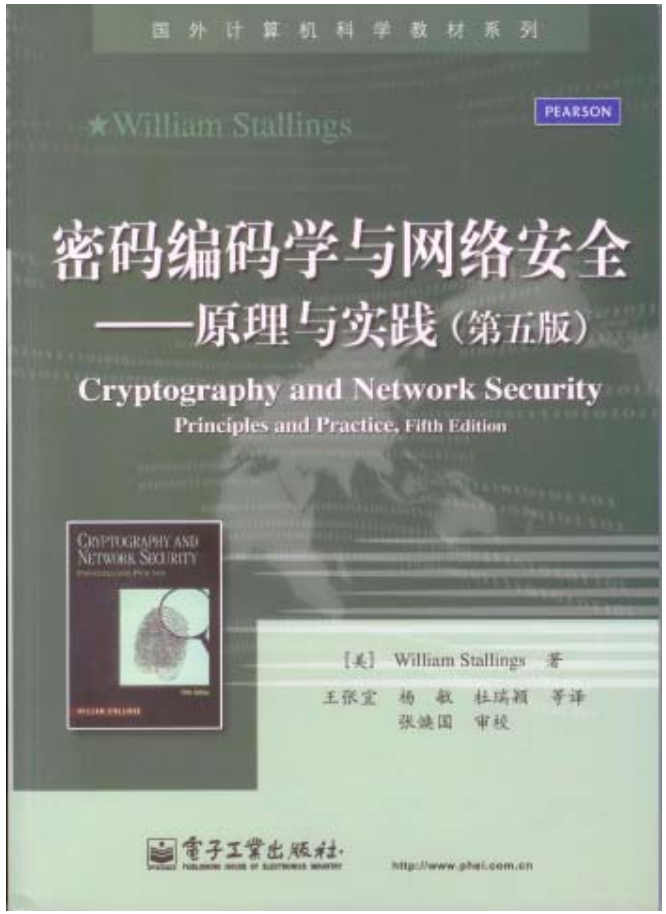
第十讲 公钥密码（2）
第十一讲 数字签名（1）
第十二讲 数字签名（2）
第十三讲 HASH函数
第十四讲 认证
第十五讲 密码协议
第十六讲 密钥管理（1）
第十七讲 密钥管理（2）
第十八讲 复习

教材与主要参考书

教材



参考书



一、公钥密码的基本思想

1、传统密码的优缺点：

①优点

- 理论与实践都很成熟。
- 安全容易把握。
- 加解密速度快。

②缺点

- 收发双方持有相同密钥，$K_e = K_d$，密钥分配困难，网络环境更突出。
- 不能方便地实现数字签名，商业等应用不方便。

一、公钥密码的基本思想

2、公开密钥密码的基本思想：

①将密钥 K一分为二：K_e 和 K_d 。K_e专门加密， K_d专门解密， $K_e \neq K_d$ 。

②由K_e 不能计算出 K_d ，于是可将K_e公开，使密钥K_e分配简单。

③由于$K_e \neq K_d$且由K_e 不能计算出 K_d ，所以 K_d 便成为用户的指纹，于是可方便地实现数字签名。

- 称上述密码为公开密钥密码，简称为公钥密码。

一、公钥密码的基本思想

3、公开密钥密码的基本条件：

①E和 D互逆； ——保密条件

$$D(E(M))=M$$

② $K_e \neq K_d$且由K_e不能计算出 K_d ； ——安全条件

③E和 D都高效； ——实用条件

④$E(D(M))=M$ ——保真条件

- 如果满足① ② ③可用于保密，如果满足② ③ ④可用于保真，如果4个条件都满足，可同时用于保密和保真。

一、公钥密码的基本思想

4、公钥密码的理论模型

(1)单向函数

设函数 $y=f(x)$，如果满足以下两个条件，则称为单向函数：

① 如果对于给定的 x，要计算出 $y=f(x)$很容易；

② 而对于给定的 y，要计算出 $x=f^{-1}(y)$很难。

(2)利用单向函数构造密码

- 用正变换作加密，加密效率高；
- 用逆变换作解密，安全，敌手不可破译；
- 但是合法收信者也无法解密。

一、公钥密码的基本思想

(3) 单向陷门函数

设函数 $y=f(x)$，且 f 具有陷门，如果满足以下两个条件，则称为单向陷门函数：

① 如果对于给定的 x，要计算出 $y=f(x)$ 很容易；

② 而对于给定的 y，如果不掌握陷门要计算出$x=f^{-1}(y)$很难，而如果掌握陷门要计算出 $x=f^{-1}(y)$就很容易。

(4) 利用单向陷门函数构造密码

① 用正变换作加密，加密效率高；

② 用逆变换作解密，安全；

③ 把陷门信息作为密钥，且只分配给合法用户。确保合法用户能够方便地解密，而非法用户不能破译。

一、公钥密码的基本思想

(5)单向函数的研究现状

- 理论上：尚不能证明单向函数一定存在；
- 实际上：密码学认为只要函数单向性足够应用就行了；
- 已找到一些单向性足够的函数：

①大合数的因子分解问题

大素数的乘积容易计算（$p\times q \Rightarrow n$），而大合数的因子分解困难($n \Rightarrow p\times q$)。

②有限域上的离散对数问题

有限域上大素数的幂乘容易计算（$a^b \Rightarrow c$），而对数计算困难($\log_a c \Rightarrow b$)。

③椭圆曲线离散对数问题

设d是正整数，G是解点群的基点，计算$dG=Q$是容易的，而由Q求出d是困难的。

二、公钥密码的基本工作方式

- 设M为明文，C为密文，E为加密算法，D为解密算法。
- 每个用户都配置一对密钥:K_e为公开的加密钥，K_d为保密的解密钥。
- 将所有用户的公开的加密钥K_e存入共享的密钥库**PKDB**。
- 保密的解密钥K_d由用户妥善保管。

PKDB

A	K_{eA}
B	K_{eB}

二、公钥密码的基本工作方式

1、确保数据秘密性：A $\xrightarrow{M}$ B

发方：

①A首先查PKDB，查到B的公开的加密钥K_{eB}。

②A用K_{eB}加密M得到密文C：$C=E(M, K_{eB})$

③A发C给B。

收方：

①B接收C。

②B用自己的K_{dB}解密，得到明文$M=D(C, K_{dB})$
$=D(E(M, K_{eB}), K_{dB})$。

二、公钥密码的基本工作方式

1、确保数据秘密性：

安全性分析：

①只有B才有K_{dB}，因此只有B才能解密，所以确保了数据的秘密性。

②任何人都可查PKDB得到B的K_{eB}，所以任何人都可冒充A给B发送数据。不能确保数据的真实性。

二、公钥密码的基本工作方式

2、确保数据真实性：A $\xrightarrow{M}$ B

发方：

①A首先用自己的K_{dA}对M解密，得到C=D(M，K_{dA})。

② A发C给B。

收方：

①B接收C。

②B查PKDB查到A的公开的加密钥K_{eA}。

③B用K_{eA}加密C，得到明文M=E（C，K_{eA}）

= E（D(M，K_{dA})，K_{eA}）。

二、公钥密码的基本工作方式

2、确保数据真实性：

安全性分析：

①只有A才有K_{dA}，因此只有A才能解密产生C，所以确保了数据的真实性。

②任何人都可查PKDB得到A的K_{eA}，所以任何人都可加密得到明文。不能确保数据的秘密性。

二、公钥密码的基本工作方式

3、同时确保数据秘密性和真实性： $A \xrightarrow{M} B$

发方：

① A首先用自己的K_{dA}对M解密，得到S：

$$S=D(M，K_{dA})$$

② A查PKDB，查到B的公开的加密钥K_{eB}。

③ A用K_{eB}加密S得到C：

$$C=E（S，K_{eB}）$$

④A发C给B。

二、公钥密码的基本工作方式

3、同时确保数据秘密性和真实性：

收方：

①B接收C。

②B用自己的K_{dB}解密C，得到S：

S=D（C，K_{dB}）

③B查PKDB，查到A的公开的加密钥K_{eA}。

④B用A的公开的加密钥K_{eA}加密S，得到M：

M=E（S，K_{eA}）

二、公钥密码的基本工作方式

3、同时确保数据秘密性和真实性：

安全性分析：

①只有A才有K_{dA}，因此只有A才能解密产生S，所以确保了数据的真实性。

②只有B才有K_{dB}，因此只有B才能获得明文，所以确保了数据的秘密性。

三、RSA公钥密码

- 1978年美国麻省理工学院的三名密码学者R.L.Rivest,A.Shamir和L.Adleman提出了一种基于大合数因子分解困难性的公开密钥密码，简称为RSA密码。
- RSA密码被誉为是一种风格幽雅的公开密钥密码。既可用于加密，又可用于数字签名，安全、易懂。
- RSA密码已成为目前应用最广泛的公开密钥密码之一。

三、RSA公钥密码

1、加解密算法

①随机地选择两个大素数 p和 q，而且保密；

②计算$n=pq$，将 n公开；

③计算 $\phi(n)=(p-1)(q-1)$，对 $\phi(n)$保密；

④随机地选取一个正整数e，$1< e <\Phi(n)$且（e , $\phi(n)$）=1，将 e 公开；

⑤根据 $ed=1 \mod \phi(n)$，求出d，并对d 保密；

⑥加密运算：$C=M^e \mod n$

⑦解密运算：$M=C^d \mod n$

● 公开加密钥$K_e=<e,n>$，保密解密钥$K_d=<p,q,d,\phi(n)>$

三、RSA公钥密码

2、算法论证

① *E*和*D*的可逆性

要证明：$D(E(M))=M$

$$M=C^d=(M^e)^d=M^{ed} \bmod n$$

因为$ed=1 \bmod \phi(n)$，这说明$ed=t\,\phi(n)+1$,其中t为某整数。所以，

$$M^{ed}=M^{t\,\phi(n)+1} \bmod n \text{ 。}$$

因此要证明 $M^{ed}=M \bmod n$，只需证明

$$M^{t\,\phi(n)+1}=M \bmod n \text{ 。}$$

三、RSA公钥密码

2、算法论证

① *E*和*D*的可逆性

在（M，n）＝1的情况下，根据数论(Euler定理)，

$$M^{t\,\phi(n)} = 1 \mod n ,$$

于是有，

$$M^{t\,\phi(n)+1} = M \mod n 。$$

三、RSA公钥密码

2、算法论证

① E和D的可逆性

注意：因为是mod n运算，所以$M \in \{0,1,2,3,\ldots,n-1\}$

在（M，n）$\neq 1$的情况下，分两种情况：

第一种情况：$M \in \{1,2,3,\ldots,n-1\}$

因为$n=pq$，p和q为素数，$M \in \{1,2,3,\ldots,n-1\}$，且（$M$，$n$）$\neq 1$。

这说明M必含p或q之一为其因子，而且不能同时包两者，否则将有$M \geq n$，与$M \in \{1,2,3,\ldots,n-1\}$矛盾。

三、RSA公钥密码

2、算法论证

① *E*和*D*的可逆性

不妨设$M=ap$。

又因q为素数，且M不包含q，故有（M，q）$=1$，是有，$M^{\phi(q)}=1 \mod q$。

进一步有，$M^{t(p-1)\phi(q)}=1 \mod q$。

因为q是素数，$\phi(q)=(q-1)$，所以$t(p-1)\phi(q)=t\phi(n)$，所以有

$$M^{t\phi(n)}=1 \mod q。$$

三、RSA公钥密码

2、算法论证

① *E*和*D*的可逆性

于是，$M^{t\phi(n)}=bq+1$，其中b为某整数。

两边同乘M，

$$M^{t\phi(n)+1}=bqM+M \text{ 。}$$

因为$M=ap$，故

$$M^{t\phi(n)+1}=bqap+M=abn+M \text{ 。}$$

取模n得，

$$M^{\phi(n)+1}=M \bmod n \text{ 。}$$

三、RSA公钥密码

2、算法论证

① E和D的可逆性

在（M，n）≠1的情况下，分两种情况：

第二种情况：$M=0$

当$M=0$时，直接验证，可知命题成立。

三、RSA公钥密码

2、算法论证

②加密和解密运算的可交换性

$$D(E(M))=(M^e)^d=M^{ed}=(M^d)^e=E(D(M)) \bmod n$$

所以，RSA密码可同时确保数据的秘密性和数据的真实性。

③加解密算法的有效性

RSA密码的加解密运算是模幂运算，运算是比较有效的。

三、RSA公钥密码

2、算法论证

④在计算上由公开的加密钥不能求出解密钥

小合数的因子分解是容易的，然而大合数的因子分解却是十分困难的。关于大合数的因子分解的时间复杂度下限目前尚没有一般的结果，迄今为止的各种因子分解算法提示人们这一时间下限将不低于

$$O(EXP(lnNlnlnN)^{1/2})。$$

根据这一结论，只要合数足够大，进行因子分解是相当困难的。

三、RSA公钥密码

2、算法论证

④在计算上由公开的加密钥不能求出解密钥

假设攻击者截获了密文C，想求出明文M。他知道

$$M \equiv C^d \mod n ,$$

因为n是公开的，要从C中求出明文M，必须先求出d，而d是保密的。但他知道，

$$ed \equiv 1 \mod \phi(n),$$

e是公开的，要从中求出d，必须先求出 $\phi(n)$，而 $\Phi(n)$ 是保密的。

三、RSA公钥密码

2、算法论证

④在计算上由公开密钥不能求出解密钥

但他又知道，

$$\phi(n)=(p-1)(q-1),$$

要从中求出 $\phi(n)$，必须先求出p和q，而p和q是保密的。但他知道，

$$n=pq,$$

要从n求出p和q，只有对n进行因子分解。而当n足够大时，这是很困难的。

三、RSA公钥密码

2、算法论证

④在计算上由公开的加密钥不能求出解密钥

只要能对***n***进行因子分解，便可攻破**RSA**密码。由此可以得出，**破译RSA密码的困难性≤对*n*因子分解的困难性。**目前尚不能证明两者是否能确切相等，因为不能确知除了对***n***进行因子分解的方法外，是否还有别的更简捷的破译方法。

四、RSA公钥密码的实现

1、参数选择

为了确保RSA密码的安全，必须认真选择密码参数：

① p和q要足够大；

- 一般应用：*p*和*q*应 512 *b* ，使*n* 达1024 *b*
- 重要应用：*p*和*q*应 1024 *b* ，使*n* 达2048 *b*

②p和q应为强素数

文献指出，只要(*p*-1)、(*p*+1)、(*q*-1)、(*q*+1)四个数之一只有小的素因子，*n*就容易分解。

③ p和q的差要大；

四、RSA公钥密码的实现

1、参数选择

④（p-1）和（q-1）的最大公因子要小。

如果（p-1）和（q-1）的最大公因子太大，则易受迭代加密攻击。

⑤ e 的选择

随机且含1多就安全，但加密速度慢。于是，有的学者建议取$e=2^{16}+1=65537$，它是素数，且二进制表示中只含两个1。

⑥ d 的选择

d不能太小，要足够大，否则不安全

⑦ 不要许多用户共用一个模 n；易受共模攻击

四、RSA公钥密码的实现

2、大素数的产生

①概率产生

目前最常用的概率性算法是**Miller**检验算法。**Miller**检验算法已经成为美国的国家标准。

②确定性产生

- 确定性测试
- 确定性构造

$E_{(G^{pub},t)}$

四、RSA公钥密码的实现

3、大素数的运算

①快速乘方算法

- 反复平方乘算法：

  设e的二进制表示为

$$e=e_{k-1}\,2^{k-1}+e_{k-2}\,2^{k-2}+\ldots,+e_1\,2^1+e_0$$

则 $$M^e=((\ldots(M^{e_{k-1}})^2M^{e_{k-2}})^2\ldots M^{e_1})^2M^{e_0}\bmod n$$

设e为k位二进制数，$w(e)$为e的二进制系数中为1的个数，则最多只需要计算$w(e)-1$次平方和$w(e)$次数的模乘。从而大大简化了计算。

四、RSA公钥密码的实现

3、大素数的运算

②快速模乘算法

- 反复平方乘算法解决了快速乘方取模的问题，仍未完全解决快速模乘的问题；
- **Montgomery**算法是一种快速模乘的好算法；
- 将以上两种算法结合成为实现**RSA**密码的有效方法。
- 硬件协处理器是提高运算效率的有效方法。

四、RSA公钥密码的实现

3、大素数的运算

- **Montgomery算法的思路：**
  - 要计算 $Y=AB \bmod n$,因为n很大，取模运算困难，采取一个小的模 R，回避大模的计算。
  - 利用空间换时间，多用存储空间换取快速。
  - 缺点：不能直接计算出 $Y=AB \bmod n$ ，只能计算出中间值 $ABR^{-1} \bmod n$ ，因此还需要预处理和调整运算。一次性计算$Y=AB \bmod n$并不划算。
  - 适合：RSA等密码中多次的模乘计算。

作 业 题

1、p165第3题,第5题。

谢　谢！