> 教材第九章，第十章，第十四章第三节

公钥密码系统原理

1976, the concept of public-key cryptographyis developed by Diffie and Hellman.

- 对称加密比较慢，公钥加密更慢。往往用对称加密保护明文，用公钥加密保护对称密钥
- 公钥加密可解决「不可否认性」问题
- 用途：加/解密（实现数据保密性），**数字签名**（实现认证），**密钥交换**（不是所有的公钥算法都有这三个用途）
- 公钥一般用于加密，验证签名；私钥用于解密，创建签名
- 公钥密码的一些误解和事实：

Misconceptions

- public-key encryption is more secure from cryptanalysis than is symmetric encryption
- public-key encryption is a general-purpose technique that has made symmetric encryption obsolete(过时)
- Public-key distribution is easy compared to secret key distribution

Facts

- security of any encryption scheme depends on the length of the key and the computational cost involved in breaking a cipher
- symmetric encryption will not be abandoned and public-key cryptography is used for key management and signature applications.
- authenticity of distributed public key should be assured

应确保公钥的真实性

- 公钥密码体制的应用

算法	加/解密	数字签名	密钥交换
RSA	✔	✔	✔
椭圆曲线	✔	✔	✔
Diffie-Hellman	✘	✘	✔
DSS	✘	✔	✘

- **对公钥密码的要求（三个容易两个不可行）：**
  1. B 产生一对密钥（公钥 PU_b，私钥 PR_b）在计算上是容易的
  2. 已知公钥和要加密的消息 M，发送方 A 产生相应的密文在计算上是容易的 $C = \mathrm{E}\left(\mathrm{PU}_b, M\right)$
  3. 接收方 B 使用其私钥对接收的密文解密以恢复明文在计算上是容易的 $M = \mathrm{D}\left(\mathrm{PR}_b, C\right) = \mathrm{D}\left[\mathrm{PR}_b, \mathrm{E}\left(\mathrm{PU}_b, M\right)\right]$
  4. 已知公钥 PU_b 时，攻击者要确定私钥 PR_b 在计算上是不可行的
  5. 已知公钥 PU_b 和密文 C，攻击者要恢复明文 M 在计算上是不可行的

RSA 算法

需要了解：

1. 公钥私钥对怎么生成，加解密怎么做
2. 数学难题是什么
3. 如何提升速率
4. 安全性

公私钥对的生成

（1）选择两个素数，$p=17$，$q=11$。
（2）计算 $n=pq=17\times11=187$。
（3）计算 $\phi(n)=(p-1)(q-1)=16\times10=160$。
（4）选择 e 使其与 $\phi(n)=160$ 互素且小于 $\phi(n)$，这里选择 $e=7$。
（5）确定 d 使得 $de \equiv 1 \bmod 160$ 且 $d<160$。因为 $23\times7=161=1\times160+1$，所以 $d=23$。d 可利用扩展的 Euclid 算法来计算（参见第 2 章）。

所得的公钥 PU = {7,187}，私钥 PR = {23,187}。加密时，需计算 $C=88^7 \bmod 187$

- 明文以分组为单位进行加密。在实际应用中，分组的大小是 i 位，其中 $2^i < n \leq 2^{i+1}$

数学难题

☆ 生成大素数是容易的，但是给定一个大数（两个素数的乘积），找出它的因子是困难的。

计算提速

- 模算术里的求幂运算：使用快速幂取模提速
- 选择特定的 e
- 使用中国剩余定理加快 $M = C^d \bmod n$ 的运算速度（教材 P204，未掌握）
- 使用扩展的欧几里得算法计算 d

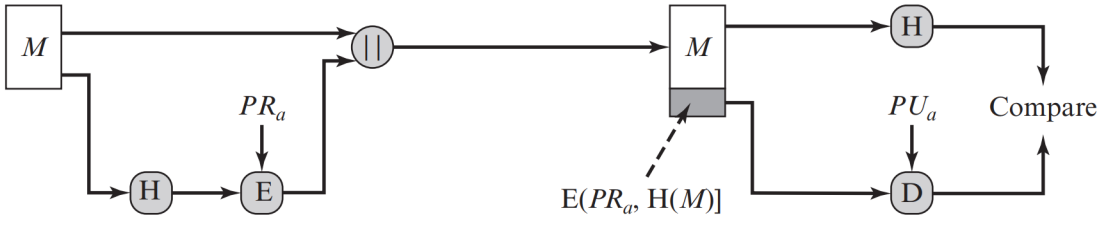
安全性

对 RSA 算法的攻击可能有如下 5 种方式：

- **穷举攻击** 这种方法试图穷举所有可能的私钥。
- **数学攻击** 有多种数学攻击方法，它们的实质都是试图分解两个素数的乘积。
- **计时攻击** 这类方法依赖于解密算法的运行时间。
- **基于硬件故障的攻击** 这种方法应用产生签名过程中处理器发生的故障。
- **选择密文攻击** 这种攻击利用了 RSA 算法的性质。

- 一般通过将因子分解的性能作为基准来评价 RSA 的安全性
- 计时攻击类似于窃贼通过观察他人转动保险柜拨号盘的时间长短来猜测密码
- 选择密文攻击见教材 P208

> [!TIP|label:当 RSA 用于数字签名时]



(a) RSA approach

密钥管理

> 第十四章第三节

人们已经提出的集中公钥分配方法本质上都可以归结为以下几种方法:

- 公开发布
- 公开可访问目录
- 公钥授权：公钥管理员会成为系统的瓶颈；管理员维护的姓名和公钥的目录也易被篡改
- 公钥证书：无需实时访问公钥授权

椭圆曲线密码

!> 椭圆曲线不会考，但是怎么做加法、做乘法需要理解

ElGamal 密钥体制

公开全局量	
q	素数
α	$\alpha < q$且α是q的素根

Alice 生成的密钥	
选择私钥 X_A	$X_A < q-1$
计算 Y_A	$Y_A = \alpha^{X_A} \bmod q$
公开密钥	$\{q, \alpha, Y_A\}$
私钥	X_A

Bob 用 Alice 的公钥加密	
明文	$M < q$
随机选择整数 k	$k < q$
计算 K	$K = (Y_A)^k \bmod q$
计算 C_1	$C_1 = \alpha^k \bmod q$
计算 C_2	$C_2 = KM \bmod q$
密文	(C_1, C_2)

用 Alicce 的私钥解密	
密文	(C_1, C_2)
计算 K	$K = (C_1)^{X_A} \bmod q$
明文	$M = (C_2 K^{-1}) \bmod q$

图 10.3　ElGamal 密码体制

例如，对于素域 GF(19)，即 q=19。素根有{2，3，10，13，14，15}。如表 2.7 所示，选择 $\alpha = 10$。Alice 生成如下的密钥对：

（1）Alice 选择 $X_A = 5$。

（2）计算 $Y_A = \alpha^{X_A} \bmod q = \alpha^5 \bmod 19 = 3$（参见表 2.7）。

（3）Alice 的私钥为 5；公钥为$\{q, \alpha, Y_A\}$={19,10,3}。

假如 Bob 想将值 M=17 发送，则

（1）Bob 选择 k=6。

（2）计算 $K = (Y_A)^k \bmod q = 3^6 \bmod 19 = 729 \bmod 19 = 7$。

（3）因此

$$C_1 = \alpha^k \bmod q = \alpha^6 \bmod 19 = 11$$
$$C_2 = KM \bmod q = 7 \times 17 \bmod 19 = 119 \bmod 19 = 5$$

（4）Bob 发送密文(11, 5)。

解密：

（1）Alice 计算 $K = (C_1)^{X_A} \bmod q = 11^5 \bmod 19 = 161051 \bmod 19 = 7$。

（2）在 GF(19)中 K^{-1} 为 $7^{-1} \bmod 19 = 11$。

（3）最终 $M = (C_2 K^{-1}) \bmod q = 5 \times 11 \bmod 19 = 55 \bmod 19 = 17$。

- 如果信息必须分组然后以加密的密钥块序列发送，那么每个分块要有唯一的 k
- 解密过程中用到了「分数取模」：$a^{p-2} \bmod p = a^{-1} \bmod p$（费马小定理或扩展欧几里得算法。费马小定理只适用于 p 为素数，扩展欧几里得算法在教材 2.3.6）
- ECC 相较于 RSA 有性能上的提升：https://zhuanlan.zhihu.com/p/46143317

思考题

9.1

> 公钥密码体制的主要成分是什么？

公钥、私钥、加密算法、解密算法、明文、密文。

9.2

> 公钥和私钥的作用是什么？

私钥可用于加密签名，任何人都可以使用公钥验证签名。公钥可用于加密只能由私钥所有者解密的信息。

9.3

> 公钥密码体制的三种应用是什么？

加/解密，数字签名，密钥交换。

9.4

> 为得到安全算法，公钥密码体制应满足哪些要求？

三个容易两个不可行。在上方 👆。

9.5 & 9.6

> 什么是单向函数？

单向函数是将一个定义域映射到一个范围内，使每个函数值都有唯一的逆，条件是函数的计算很简单，而逆的计算是不可行的。

> 什么是单向陷门函数？

单向陷门函数在一个方向上容易计算，而在另一个方向上不可行，除非已知某些附加信息。有了额外的信息，逆可以在多项式时间内计算。

10.1

> 简要说明 Diffie-Hillman 密钥交换。

双方各自创建公私钥对，并将公钥传送给对方。密钥的设计使得双方可以根据各自的私钥和对方的公钥计算出相同的唯一密钥。（它是一种建立秘钥的方法，而不是加密方法）

习题

9.11

> “我想告诉你，福尔摩斯，”华生激动地说，“你最近进行的网络安全活动让我对密码学产生了浓厚的兴趣，就在昨天，我发现一次一密的加密方法是可行的。”
> “噢？真的吗？”福尔摩斯从睡意蒙眬中醒过来。“这么说，你找到了一种生成强密码序列的确定性方法了？”
> “千真万确，福尔摩斯。这个想法倒是挺简单的。对给定的单向函数 F，通过将 F 应用于某标准的变量参数序列，我就产生了一个长伪随机数序列。假使密码分析者知道了 F 和序列的一般性质，这个性质可能很简单，比如 S，S+1，S+2，...，而不知道 S，由于 F 的单向性，没人能够对某 i，从 $F(S+i)$推出 S，即使他得到了序列的一段，他也不能确定其他部分。”
> “华生，我担心你的想法并非无懈可击，至少它要求 F 满足一些附加条件。我们考虑一下。例如，RSA 加密函数 $F(M)=M^K \bmod N$，K 是保密的，这个函数被认为是单向的，但是我不赞成将这种方法用于类似 M=2，3，4，5，6，…这样的序列。”
> “为什么，福尔摩斯？”华生大惑不解，“为什么你认为，如果 K 是保密的，那么像 $2^K \bmod N$, $3^K \bmod N$, $4^K \bmod N$, …这样的序列不适合于一次一密？”

因为第三项等于第一项的平方；第五项等于第一项乘以第二项.....

9.15

> “这是一个非常有趣的案例，华生。”福尔摩斯说，“这个年轻人爱上了一个女孩，这个女孩也爱他。但是女孩的父亲非常怪，他坚持要求他未来的女婿以公钥密码体制为基础设计一个简单安全的协议，以便他在公司的计算机网络中使用。这个年轻人提出了下列两方通信协议：假设用户 A 要将消息 M 发送给用户 B［交换的消息形为(发送方的姓名，消息正文，接收方的姓名)］”
> （1）A 将$(A, \mathrm{E}(\mathrm{PU}_b, [M, A]), B)$发送给 B；
> （2）B 发送应答$(B, \mathrm{E}(\mathrm{PU}_a, [M, B]), A)$给 A；
> “这个协议确实很简单，但是女孩的父亲还是认为该协议不够简单，因为这种协议中存在一些冗余，可进一步简化为
> （1）A 将$(A, \mathrm{E}(PU_b, M), B)$发送给 B；
> （2）B 发送应答$(B, \mathrm{E}(PU_a, M), A)$给 A；
> 由于这个原因，女孩的父亲不许他的女儿与年轻人结婚，这使得他们非常不愉快，因此年轻人来我这里请求我的帮助。”
> “嗯，我不知道你会怎样帮助他。”华生想到年轻人要失去他心爱的人，显得有些不快。
> “我想我可以帮助他，你知道，华生，冗余有时候对保证协议的安全性是有好处的，因此女孩父亲简化后的协议容易受到一种攻击，而年轻人设计的协议能够抗这种攻击。”福尔摩斯若有所思地说，“有办法了，华生。瞧，攻击者必须是网络用户中的一员，且能够截获 A 和 B 交换的消息。因为是网络中的用户，所以他自己也有公钥，并且他可以发消息给 A 或 B，也可以接收 A 或 B 发出的消息。如果使用这个简化后的协议，那么他可以按下述过程得出 A 以前发送给 B 的消息 M，……”
> 请完成上述的过程。

1. 敌方 X 拦截了 A 发给 B 的消息，$[A, \mathrm{E}(PU_b, M), B]$
2. X 给 B 发送 $[X, \mathrm{E}(PU_b, M), B]$
3. B 通过给 X 发送 $[B, \mathrm{E}(PU_x, M), X]$ 表示 ta 收到了 X 发送的消息
4. X 使用他的私钥解密 $\mathrm{E}(PU_x, M)$，从而得到 M

9.18

> 这个习题说明了选择密文攻击的简单应用。Bob 截获了一份发给 Alice 的密文 C，该密文是用 Alice 的公钥 e 加密的。Bob 想获得原始消息 $M=C^d \bmod n$。Bob 选择一个小于 n 的随机数 r，并计算
>
> $$Z = r^e \bmod n$$
> $$X = ZC \bmod n$$
> $$t = r^{-1} \bmod n$$
>
> 接着，Bob 让 Alice 用她的私钥对 X 进行认证（如图 9.3 所示），从而对 X 进行解密。Alice 返回 $Y=X^d \bmod n$。请说明 Bob 如何利用获得的信息去求取 M。

e 是公钥，d 是私钥。

因为 $Z = r^e \bmod n$，所以 $r = Z^d \bmod n$。Bob 可以算出：

$tY \bmod n = r^{-1}X^d \bmod n = r^{-1}Z^dC^d \bmod n = C^d \bmod n = M$