# Euler Function and Fermat Little Theorem

# 欧拉函数及费尔马小定理

# Euler Function Φ(n)—Euler Theorem

对于任意的大于1的正整数n，关注这样的集合：{m|gcd(m,n)=1, m为小于等于n的正整数}， 也即 1到n范围内与n互素的数的集合

**欧拉函数定义：** (The Euler Φ function) 记Φ(n)为1到n之间与n互素的整数的个数。也即满足 1 ≤ k ≤ n, 且gcd(k, n) = 1 的正整数的个数。

欧拉定理，欧拉函数，以及这个集合在计算机安全方面有着重要的应用

It is the basis for a certain type of encryption known as RSA (discussed below) and is used in a common encryption protocol called **PGP** (Pretty Good Privacy).

# 欧拉函数Euler Function Φ(n)—important

**欧拉函数定义** (The Euler Φ function)：记Φ(n)为1到n之间与n互素的整数的个数。也即满足 1 ≤ k ≤ n, 且gcd(k, n) = 1 的正整数的个数。

**Example** Φ(12) = 4. （{1,5,7,11}）
Since gcd(1, 1) = 1, we have Φ(1) = 1.
当n很大时，欧拉函数值很难计算，但如果n是素数呢？

**Property of the Euler Φ function** : For any prime p, we have Φ(p) = p − 1. （重要性质）

Why?

because gcd(k, p) = 1 for k = 1, 2, . . . , p − 1.

# Euler Function—calculate Φ(pq)

**重要性质：** p,q为两个不同的素数，n = pq 那么n的欧拉函数值

$$\Phi(pq) = pq - (p + q - 1) = (p - 1)(q - 1).$$

**Important! RSA will use this property**

**事实上，有如下性质：**

如果gcd(m,n)=1，那么**Φ(mn) = Φ(m) Φ(n)**

利用容斥原理可证

# Euler Function Φ(n)—Euler Theorem

- **Euler Theorem 欧拉定理**: 对任意大于1的正整数n，有 $a^{\Phi(n)} \equiv 1 \pmod{n}$,其中 gcd(a,n)=1 (一个与n互素的整数 )
- **Example:** Suppose n = 12. We know that Φ(12) = 4 and that the units are {1, 5, 7, 11}.
- 验证： 显然 $1^{\Phi(12)} \equiv 1 \bmod 12$),
  - $5^4 = (24+1)^2 \equiv 1 \pmod{12}$.
  - $7^4 = (12-5)^4 \equiv 5^4 \pmod{12} \equiv 1 \pmod{12}$.
  - Likewise, $11 \equiv -1 \pmod{12}$ and so $11^4 = (-1)^4 \equiv 1 \pmod{12}$.

- 结论： If n = pq then, since Φ(n) = (p – 1)(q – 1), this property becomes $m^{(p-1)(q-1)} \equiv 1 \pmod{pq}$ when gcd(m, pq) = 1 and p,q are primes.

Pierre de Fermat
(1601-1665)

# Fermat Little Theorem 费尔马小定理

**Fermat Little Theorem**费尔马小定理: **如果p是素数**,对于任一个与p互素的整数a, 都有：

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

（或者说a不是p的倍数？）

- 另一种形式是, 设$p$是素数, 则对任意的整数$a$,

$$a^p \equiv a \pmod p.$$ (实际上可以说是欧拉定理的一个推论）

***These two properties will play an important role in our discussion of the RSA protocol.***

费尔马大定理:对任何正整数a,b,c和n,当n>2时,$a^n+b^n \neq c^n$. *(从猜想到证明花了几个世纪，顺便介绍)*

# 优美而有力的结论

- **欧拉定理:** $a^{\Phi(n)} \equiv 1 \pmod{n}$ where $\gcd(a,n)=1$

- **费尔马小定理 Fermat Little Theorem：** $a^{p-1}=1 \pmod{p}$ 其中，p是素数，a不是p的倍数

# 费尔马小定理应用举例

- 例题：利用费尔马小定理计算 $7^{228} \bmod 11$
- 解 (充分利用模运算的性质和费尔马小定理，简化计算，也让同学们体会如何简化模余运算)
- $7^{228} \bmod 11 = [(7^{220} \bmod 11)*(7^{8} \bmod 11)] \bmod 11$

  $= [(7^{10})^{22} \bmod 11)*((11-4)^{8} \bmod 11)] \bmod 11$ (应用费尔马小定理)

  $=[(7^{10})^{22} \bmod 11)*((11-4)^{8} \bmod 11)] \bmod 11$

  $=1*(-4)^{8} \bmod 11 = 2^{16} \bmod 11 = [(2^{10} \bmod 11)* 2^{6} \bmod 11] \bmod 11$

  $=1*2^{6} \bmod 11 = 9$ (应用费尔马小定理)

  $= [(2^{5} \bmod 11)\ 2 \bmod 11] \bmod 11 = \{[(-1) \bmod 11] *2 \bmod 11\} \bmod 11$

  $= (-2) \bmod 11 = (9-11) \bmod 11 = 9$

以上这些过程是为了让大家熟悉模运算和费尔马小定理的应用

# 费尔马小定理--中国余数定理

## 综合应用

- 例题：假设p,q是不同素数，证明 $p^{q-1}+q^{p-1} = 1 \bmod pq$
- 解法1 记$p^{q-1}+q^{p-1}$ ,由$a=1 \pmod p$, 而且 $a=1 \pmod q$知，存在两个整数s,t使得$a = sp+1$, $a=tq + 1$. 于是 $sp = tq$. 由于p,q互素，所以p|t, q|s. 于是pq|( a -1 ). 所有 $a = 1 \pmod{pq}$

- 解法2提示：(利用中国余数定理)
- 记$a= p^{q-1}+q^{p-1}$ , 那么由费尔马小定理知,
  - p,q 为不同素数，那么gcd(p,q)=1. $a=1 \pmod p$, 而且 $a=1 \pmod q$。
  - 所以a是同余方程组$\begin{cases} x = 1\ (mod p) \\ x = 1\ (mod q) \end{cases}$ 的一个解。 显然整数1是该同余方程组在0到pq之间的唯一的解。
  - 由中国余数定理知，该同余方程组的解都关于pq模同余
  - 于是1 与$p^{q-1}+q^{p-1}$ 关于模pq同余。

# 费马小定理应用举例--伪素数

- **结论：**对于一个给定的数n>2, 如果n是素数，那么gcd(2, n)=1. 于是由费马小定理得到 $2^{(n-1)} \equiv 1 \bmod(n)$.

- **思考：**如果$2^{(n-1)} \equiv 1 \bmod(n)$不成立，能得到什么结论？

费马小定理提供了一种不用因子分解就能断定一个数是合数的新途径： 选择某个不等于n的素数p, 如果$p^{(n-1)} \equiv 1 \bmod(n)$不成立，那么n一定是合数。

**例如，** $2^{9-1} \equiv 4 \pmod 9$ （$\neq 1 \bmod(9)$ ），可以断定9是合数.

**问题：**如果$p^{(n-1)} \equiv 1 \bmod(n)$成立，是否一定有n为素数？

举例： $2^{340} \equiv 1 \bmod(n)$， 但341=11*31不是素数。

# 伪素数

- 对于寻找素数，尤其是大素数，是很有使用价值的。

- 定义（伪素数）：p>1是一个正整数，如果$p^{(n-1)} \equiv 1 \mod(n)$成立，但n不是素数，则称n为以p为基数的伪素数。
- 例如： 341=11*31 就是以2为基数的伪素数

# Carmichael number 卡米切尔数

- Definition: A composite integer $n$ that satisfies the congruence $b^{n-1} \equiv 1 \pmod{n}$ for all positive integers $b$ with $\gcd(b, n) = 1$ is called a ***Carmichael number***.

- For example: 561=3·11·17 is a ***Carmichael number***

- 这种数是素数的概率很大，也为快速寻找大素数，做素数测试提供了一种途径。

# 欧拉函数值Φ(n)的计算公式--optional

例题：假设n是大于1的正整数，则根据素数分解定理知，

$n= p_1^{a1}p_2^{a2}...p_k^{ak}$ 为n的素数分解，利用容斥原理计算正整数n的欧拉函数值。

解答：令：$A_i = \{ x|0\le x \le n-1且p_i|x\}$, $|A_i| = n/p_i$, 为什么？

$\overline{A_1}$ 代表什么??

那么：$\Phi(n)=|\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_K}|$ （代表什么含义？？）

$|A_i \cap A_j| = n/(p_i,p_j)$ , $0\le i<j \le k$

$|A_1 \cap A_2 \cap \cdots \cap A_k|=n/(p_1p_2...p_k)$

于是根据容斥原理，就有：$\Phi(n)=|\overline{A_1} \cap \overline{A_2} \cap \cdots \cap \overline{A_K}|$

$$= n - \left(\frac{n}{p_1}+\frac{n}{p_2}+...+\frac{n}{p_k}\right) + \left(\frac{n}{p_1p_2}+\frac{n}{p_1p_3}+...+\frac{n}{p_{k-1}p_k}\right) - ...+(-1)^k\frac{n}{p_1p_2\cdots p_k}$$

$$= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)...\left(1-\frac{1}{p_k}\right)$$

- 例如：$\Phi(60) = 60(1-1/2)(1-1/3)(1-1/5) = 16$

# 练习

- Edition 8th 4.4节
- T33, T38（这道题的很有意思，也给了一种求某些模余的思路）, T40，

- 选做T15

# 数论在在Cryptography密码学中的应用

# Cryptography and Secrecy—Basic Idea

明文转换成密文的规则可以说就是函数.如果用**P表示所有涉及到的可能的明文信息的集合，C表示所有的密文信息的集合；**K 是相关的密码，发送者用来加密的key.

函数 $f_K : P \rightarrow C$ (加密函数)

接收方则采用逆函数

$f_k^{-1}: C \rightarrow P$ (解密函数)

来解密。当然要求逆函数$f_k^{-1}$是存在的。

$f_k^{-1}$ 要存在， 加密函数$f_K$必须满足什么条件？

# Cryptography and Secrecy—Basic Idea

Example 简单加密举例

一个简单的位串 10101001；

K = 11000111, 加密用的key

可以采用简单的异或，将明文与key做异或运算，得到一个简单的密文

10101001 plaintect

11000111 key K

0110110 ciphertect

01101110 ciphertext

11000111 key k

10101001 plaintect

就这个操作而言，加密解密都是一样的异或操作，key也一样。

$f_k^{-1} = f_K$

显然，这是一种加密；但显然也无密可言，过于简单。

# Cryptography and Secrecy—Basic Idea

上面简单加密例子，还有两个问题需要考虑：

1. 只加密了长度为8位的位串；如果需要加密更长的信息如何？
2. 如果待加密的明文不是位串，是其它信息，如中英文等，如何？

• 所以，只要能加密整数，就可以对任何信息进行加密。后面的讨论只谈如何加密整数。

# Shift Cipher 移位加密举例

**Example**: Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with k = 11.

**Solution**: Replace each letter with the corresponding element of $\mathbf{Z}_{26}$.

18 19 14 15  6 11 14 1 0 11  22 0 17 12  8  13  6.

Apply the shift  f(p) = (p + 11) **mod** 26, yielding

3 4 25 0  17 22 25 12 11 22  7 11 2 23  19  24  17.

Translating the numbers back to letters produces the cipher text

"DEZA RWZMLW HLCXTYR."

# Cryptosystems 加密系统

**Definition**: 加密系统五要素： $(\mathcal{P},C,\mathcal{K},\mathcal{E},\mathcal{D})$:

- $\mathcal{P}$ is the set of plaintext strings,
- $C$ is the set of cipher text strings,
- $\mathcal{K}$ is the *key space* (set of all possible keys),
- $\mathcal{E}$ is the set of encryption functions, and
- $\mathcal{D}$ is the set of decryption functions.

- The encryption function in $\mathcal{E}$ corresponding to the key $k$ is denoted by $E_k$ and the decryption function in $\mathcal{D}$ that decrypts cipher text encrypted using $E_k$ is denoted by $D_k$. Therefore:

$$D_k(E_k(p)) = p, \text{ for all plaintext strings } p.$$

# 两类密码体系

- **私钥密码体系（对称加密）**：

  私钥加密算法使用单个私钥来加密和解密数据。由于具有密钥的任意一方都可以使用该密钥解密数据，因此必须保护密钥不被未经授权的代理得到。私钥加密又称为对称加密，因为同一密钥既用于加密又用于解密。私钥加密算法非常快（与公钥算法相比），特别适用于对较大的数据流执行加密转换。 如DES加密算法。

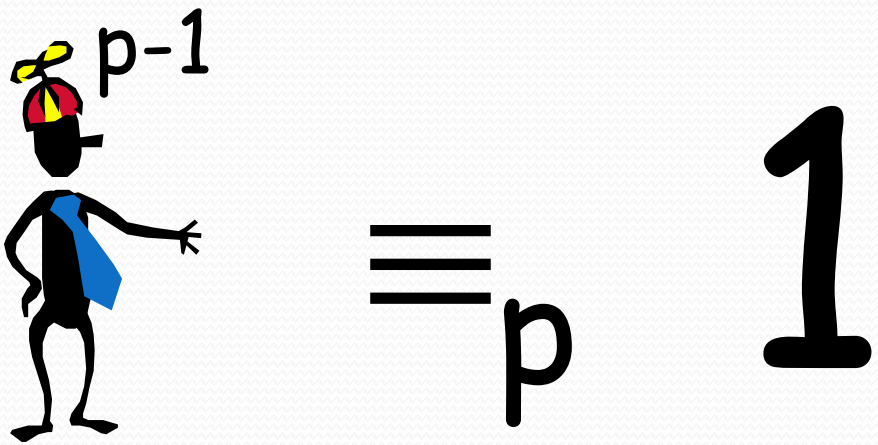
- **公钥密码体系（非对称加密）**:加密密钥公开,解密密钥保密：

  公钥加密使用一个必须对未经授权的用户保密的私钥和一个可以对任何人公开的公钥。公钥和私钥都在数学上相关联；用公钥加密的数据只能用私钥解密。如RSA加密算法

# R S A

(1978)

# Modular Arithmetic and the RSA Cryptosystem




Fermat Little Theorem

# Starring

Rivest

Shamir

Adleman

Euler

Fermat

# RSA公钥密码

- RSA是公钥密码 (R. Rivest, A. Shamir, L. Adleeman,1978)

1. 随机选取2个（足够大的）大素数$p$和$q$($p<q$), 记$n=pq$, $\phi(n)=(p-1)(q-1)$.

2. 选择正整数e, $e$与$\phi(n)$互素, 令$d=e^{-1}(\mathrm{mod}\,\phi(n))$. ($e^{-1}$为e 关于模$\phi(n)$的模逆）(这个d就是解密私钥，而e则为加密公钥)

3. 将明文数字化, 分成若干段, 使得每一个明文段$m<n$.
(最好是每一段明文m<min(p,q),当p,q足够大时没任何问题，可以保证m与n互素)

4. 加密算法(函数): $c=E(m)=m^e \bmod n$，*明文m与模n互素*

5. 解密算法(函数): $m=D(c)=c^d \bmod n$

- 其中加密公钥$e$和$n$是公开的, 而$p,q$, $\phi(n)$和$d$(*秘钥*)是保密的.

# RSA 加解密过程

| 收报方 | 公开通讯 | 发报方 |
| --- | --- | --- |
| 1.选择确定p、q、e、d | | 0.待发明文：$x_1$、$x_2$、…、$x_n$ |
| 2. 发出N、e (公钥) | N=pq、e | |
| p, q, d(秘钥)是保密的部分,$\phi(n)$当然也保密 | | 3、发出$c_i \equiv x_i^e \pmod N$<br>i=1,2,…,n |
| | 密文$c_1$、$c_2$…、$c_n$ | |
| 4、由 $c_i$解出$x_i$：<br>$x \equiv c^d \pmod N$ | | 思考两个问题：1 为什么能起到加密作用？2：为什么这个公式能解密？ |

# Cryptography –RSA

*回忆：* **欧拉定理**: $a^{\Phi(n)} = 1 \pmod n$， a与n互素

**费尔马小定理** $a^{p-1}=1 \pmod p$，p为素数，a与p互素

以下说明为什么RSA算法正确

# How does RSA's decryption method work

计算发送 密文C (= $M^e$%N ); 由于 ed = 1 (mod Φ(N)). 因此有某个k 使得 ed = 1 + k Φ(N)

所以 $C^d = (M^e)^d = M^{ed}$ (mod N )

$M^{ed} = M^{1+k\,\Phi(N)} = M \times (M^{\Phi(N)})^k$

再因为 gcd(M,N) = 1, 可以准确地恢复计算出明文M

由$M^{\Phi(N)} = 1$ mod N （**欧拉定理**）可得:

$C^d = M^{ed} = M \times (1)^k = M$ (mod N ) = M

问题：这里的明文M必须是与N互素的，否则上面推导有问题。那么如果实际上的M与N不互素，如何？

# Cryptography –RSA 举例说明

RSA 加密算法基于 **N = pq，p and q 两个不同素数**

举例：假设选择 N = 77= **pq** where p = 7, q = 11.

实际应用中：p 、q 会是数百位甚至更大的素数。 这样即使知道pq乘积的结果N， 但由于数字太大，基于当今有的技术水平，很难在有限时间内分解出来，这就是RSA算法安全有用的原因

公开：N=77,

注：假定不知道分解式7*11

# Cryptography –RSA

因为 60 = (p − 1)(q − 1) = Φ(77).
选择一对互为模逆的整数e 、d
满足 ed = 1 (mod 60).
e 与d 互为模逆（关于Φ(77)的模逆，不是关于77的模逆）

例如如果我们选择 e = 13，那么 gcd(13,60) = 1, 计算出模逆
d = 37.
e × d = 13 × 37 = 481 where 481 = 1 (mod 60).

把 e = 13 作为公钥 （公开13， 77)
把37作为密钥秘密保存 d = 37 ( Φ(77)=60 当然也是秘密 )

# Cryptography –RSA

So far, N = 77, e = 13 are public
分解式、密钥、 Φ (77) = 60 都是秘密

对于明文M满足 1<=M < 77 where gcd(M, N) = 1

例如 明文 M = 5 (plaintext).
基于公开信息 (77 and 13, public) (公开的key)
计算: $M^e \bmod 77 = 5^{13} \bmod 77 = 26$ (ciphertext 密文). (发送方发送）
$M^e\%N$ (加密公式）

收到信息的解密方：to decrypt the ciphertext C, 公式： $C^d \bmod N$
计算 $C^d \bmod N = 26^{37} \bmod 77$ gets 5(plaintext) （利用私钥d=37 )

# Cryptography –RSA key的选择

如何选择公钥与密钥 e、d?

as long as $\gcd(e, 60) = 1$

$ed=1 \pmod{60}$

$d=e^{-1} \pmod{60}$

就可以，并非只有13对37

# The problem from RSA 存在的问题

素数p、q是数百位上千位的，e、d的选择也需要注意

由公式推导知道，明文需要满足 gcd(M,N) = 1. 需要满足：明文M与模N互素

因为：

$$\phi(N)/N = (1-\frac{1}{P})(1-\frac{1}{q})$$

, 可以看出，不满足的概率很小，当N很大时。

For example, we can make p,q enough such as M < Min(p,q) 这样可以保证M与N互素，保证使用RSA算法的正确性

*注：根据习题28的结论，即使是M与pq不互素，解密也正确。*

# Cryptography –RSA的安全性

如何破解RSA? 需要知道私钥,或者 Φ (n) , 或者知道N=pq的分解式。

RSA is based on

**the difficulty of factoring of integer.**

In this case, N=pq. 当p,q足够大时，无法分解，至少在短时间内难以分解。

因为不知道p,q, 当然也不知道$\phi(n)$。 于是无法解出私钥 $d=e^{-1}(\mathrm{mod}\,\phi(n))$. 起到加密作用

Without secret keys (even with public e and N) the spy has *to factor N in order to decrypt*

*— too hard when N is big enough!*

# RSA-Key length(位数)的概念

When we talk about the *key length* of an RSA key, we are referring to **the length of the modulus N in bits**. 也即模值N的位数（作为2进制数的长度）。

The minimum recommended key length for a secure RSA transmission is currently 1024 bits.

A key length of 512 bits is now no longer considered secure, although cracking it is still not a trivial task for the likes of you and me.

The longer your information is needed to be kept secure, the longer the key you should use. Keep up to date with the latest recommendations in the security journals （跟踪最新建议）.

并非位数越长越好，为什么？

# RSA总结

- 加密函数（算法）、解密函数都是公开的，而且也很简单。
- 加密用的key也是公开的 （公钥）
- 但由于整数分解的困难，使得这种算法能达到加密的目的
- 思考：RSA的安全性基于整数分解的难度，计算速度不够快。如果将来有了好的整数分解算法，或者计算速度大幅提高，如量子计算机的使用如何？

- 实际应用过程中，还需要防止伪造的公钥，也即需要核实公钥发布者的身份。