数论基础

2020 年 12 月

本章要点

- 素数是一种整数，在整除意义下，它只能被自身（正负）和 1 整除。素数在数论和密码学里扮演重要角色。
- 在公钥密码中起重要作用的两个定理是费马定理和欧拉定理。
- 许多密码算法的一个重要前提是能够选择一个大的素数。开发有效算法判定一个随机整数是否为素数（即素性测试）是密码学研究的重要课题。
- 离散对数是许多公钥算法的基础。离散对数和普通对数类似，但是在模算术上进行运算。

主要内容

主要内容

素数 Prime Numbers

- 整数 $p > 1$ 是素数当且仅当它只有因子 ± 1 和 $\pm p$。
  - 如 $2, 3, 5, 7$ 是素数，而 $4, 6, 8, 9, 10$ 不是素数。
- 素数不能写作其他数的乘积形式。
- 1 是素数，但是通常没有什么用。
- 素数是数论的核心。

小于 2000 的素数分布情况

2	101	211	307	401	503	601	701	809	907	1009	1103	1201	1301	1409	1511	1601	1709	1801	1901
3	103	223	311	409	509	607	709	811	911	1013	1109	1213	1303	1423	1523	1607	1721	1811	1907
5	107	227	313	419	521	613	719	821	919	1019	1117	1217	1307	1427	1531	1609	1723	1823	1913
7	109	229	317	421	523	617	727	823	929	1021	1123	1223	1319	1429	1543	1613	1733	1831	1931
11	113	233	331	431	541	619	733	827	937	1031	1129	1229	1321	1433	1549	1619	1741	1847	1933
13	127	239	337	433	547	631	739	829	941	1033	1151	1231	1327	1439	1553	1621	1747	1861	1949
17	131	241	347	439	557	641	743	839	947	1039	1153	1237	1361	1447	1559	1627	1753	1867	1951
19	137	251	349	443	563	643	751	853	953	1049	1163	1249	1367	1451	1567	1637	1759	1871	1973
23	139	257	353	449	569	647	757	857	967	1051	1171	1259	1373	1453	1571	1657	1777	1873	1979
29	149	263	359	457	571	653	761	859	971	1061	1181	1277	1381	1459	1579	1663	1783	1877	1987
31	151	269	367	461	577	659	769	863	977	1063	1187	1279	1399	1471	1583	1667	1787	1879	1993
37	157	271	373	463	587	661	773	877	983	1069	1193	1283		1481	1597	1669	1789	1889	1997
41	163	277	379	467	593	673	787	881	991	1087		1289		1483		1693			1999
43	167	281	383	479	599	677	797	883	997	1091		1291		1487		1697			
47	173	283	389	487		683		887		1093		1297		1489		1699			
53	179	293	397	491		691				1097				1493					
59	181			499										1499					
61	191																		
67	193																		
71	197																		
73	199																		
79																			
83																			
89																			
97																			

每一列表示 100 个连续自然数。

合数的素因子分解

- 分解一个合数 n 就是把它写成其他整数的乘积的形式：

$$n = a \times b \times c$$

其中 a, b, c 为整数。

- 比起用乘的方法把几个因子乘起来生成整数，分解整数通常困难的多。

- 任何整数 $a > 1$，都可以唯一的分解为

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

其中 $p_1 < p_2 < \ldots < p_t$ 是素数，所有 a_i 都是正整数。

- 例如，
$91 = 7 \times 13, 3600 = 2^4 \times 3^3 \times 5^2, 11011 = 7 \times 11^2 \times 13$。

合数的素因子分解

- 素因子分解就是把一个整数写成若干个素数的乘积的形式：

$$a = \prod_{p \in P} p^{a_p}$$

其中 P 是所有素数构成的集合，$a_p \geq 0$ 是整数。

- 对于某一个整数 a，其大多数指数 a_p 为 0.
- 任一给定的正整数, 可通过简单列出所有后面公式中非零指数分量来说明。
- 例如，12 可以表示为 $\{a_2 = 2, a_3 = 1\}$，18 可以表示为 $\{a_2 = 1, a_3 = 2\}$，91 可以表示为 $\{a_7 = 1, a_{13} = 1\}$。

合数的素因子分解

- 两个数的乘法等同于对应指数分量的加法

$$k = mn \Rightarrow k_p = m_p + n_p$$

其中 p 为任意素数。

- 例如，$k = 12 \times 18 = (2^2 \times 3) \times (2 \times 3^2) = 216, a_2 = 2 + 1 = 3, a_3 = 1 + 2 = 3, 216 = 2^3 \times 3^3$.

- 任何以 p^k 形式表示的整数仅能被对应素数分量小于或等于它的另一个整数 p^j 整除, 其中 $j \leq k$，即有

$$a|b \Rightarrow a_p \leq b_p$$

其中 p 为任意素数。

- 例如，$a = 12, b = 36, 12|36, 12 = 2^2 \times 3^1, 36 = 2^2 \times 3^2, a_2 = 2 = b_2, a_3 = 1 < 2 = b_3$.

最大公约数 GCD

- $\gcd(a, b) = c$，即 c 是 a 和 b 的最大公约数，当
  - c 是 a 和 b 的因子；
  - 任何 a 和 b 的因子也是 c 的因子。

- 下列关系总是成立的：如果 $k = \gcd(a, b)$，则 $k_p = \min(a_p, b_p)$，对于任意的 $p \in P$。

- 两个整数 a, b，如果除了 1 以外没有公共因子，则称它们互素（Relatively Prime）。
  - 8 和 15 互素，因为 8 的因子是 $1, 2, 4, 8$，而 15 的因子是 $1, 3, 5, 15$，1 是它们唯一的公共因子。

- 如果将整数表示为素数之积，则容易确定两个正整数的最大公因子。
  - $300 = 2^2 \times 3^1 \times 5^2, 18 = 2^1 \times 3^2, \gcd(18, 300) = 2^1 \times 3^1 \times 5^0 = 6$

欧几里得算法（Euclidean Algorithm）

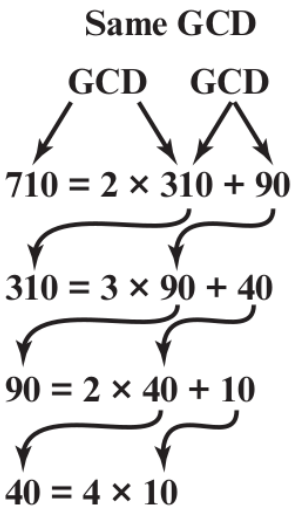
- 欧几里得（Euclid，约公元前 330 年 – 公元前 275 年），古希腊人，数学家，被称为“几何之父”。他最著名的著作《几何原本》是欧洲数学的基础，被广泛认为是历史上最成功的教科书。
  - 欧几里得算法也叫辗转相除法，首次出现于欧几里得的《几何原本》。在中国则可以追溯至东汉时出现的《九章算术》。
  - 欧几里得算法是数论中的一个最基本技巧，它可以简单的求两个正整数的最大公约数。

欧几里得算法的原理

- 假设要求整数 a 和 b 的最大公因子，不妨令 $a \geq b > 0$.
- b 除 a 可以表示为 $a = qb + r$，其中 $0 \leq r < b$ 为余数。
- 如果 $r = 0$，则 $\gcd(a, b) = b$。
- 如果 $r \neq 0$，考虑 $\gcd(a, b)$ 和 $\gcd(b, r)$ 之间的关系：
  - 令 $d = \gcd(a, b)$。因为 $d|a$ 且 $d|b$，所以 $d|(a - qb)$，即 $d|r$。也就是说，d 是 b, r 的公因子。那么，$d \leq \gcd(b, r)$。
  - 令 $c = \gcd(b, r)$。因为 $c|b$ 且 $c|r$，所以 $c|(qb + r)$，即 $c|a$。也就是说，c 是 a, b 的公因子。因为 a, b 的最大公因子是 d，所以 $c = \gcd(b, r) \leq d$。
- 所以：$\gcd(a, b) = \gcd(b, r)$，即求 a 和 b 的最大公因子可以转化为求 b 和 r 的最大公因子。

欧几里得算法的原理

- 欧几里得算法的原理可以表示为：对任意整数 a, b，且 $a \geq b > 0$，则 $\gcd(a, b) = \gcd(b, a \bmod b)$。
- 也就是说，求 a, b 的最大公约数可以转化为求 b 和 b 除 a 余数的最大公约数，即辗转相除。




```
Euclid(a, b){
  if (b=0) then return a;
  else return Euclid(b, a mod b);
}
```

扩展欧几里得算法

- 给定两个整数 a 和 b，扩展欧几里得算法不仅可以求出最大公因子 d，而且可以得到两个整数 x 和 y，满足 $ax + by = d = \gcd(a, b)$。
- 利用欧几里得算法，并且假设每步 i 都可得到 x_i 和 y_i 满足 $r_i = ax_i + by_i$。则有以下关系式

$$
\begin{aligned}
a &= q_1 b + r_1 & r_1 &= a x_1 + b y_1 \\
b &= q_2 r_1 + r_2 & r_2 &= a x_2 + b y_2 \\
r_1 &= q_3 r_2 + r_3 & r_3 &= a x_3 + b y_3 \\
&\vdots & &\vdots \\
r_{n-2} &= q_n r_{n-1} + r_n & r_n &= a x_n + b y_n \\
r_{n-1} &= q_{n+1} r_n + 0
\end{aligned}
$$

扩展欧几里得算法

Now, observe that we can rearrange terms to write

$$r_i = r_{i-2} - r_{i-1}q_i \tag{2.8}$$

Also, in rows $i - 1$ and $i - 2$, we find the values

$$r_{i-2} = ax_{i-2} + by_{i-2} \quad \text{and} \quad r_{i-1} = ax_{i-1} + by_{i-1}$$

Substituting into Equation (2.8), we have

$$\begin{aligned} r_i &= (ax_{i-2} + by_{i-2}) - (ax_{i-1} + by_{i-1})q_i \\ &= a(x_{i-2} - q_i x_{i-1}) + b(y_{i-2} - q_i y_{i-1}) \end{aligned}$$

But we have already assumed that $r_i = ax_i + by_i$. Therefore,

$$x_i = x_{i-2} - q_i x_{i-1} \quad \text{and} \quad y_i = y_{i-2} - q_i y_{i-1}$$

- 从而可以递推地得到 x_i 和 y_i。

扩展欧几里得算法

$r_0 = b$		$x_0 = 0; y_0 = 1$	$b = ax_0 + by_0$
$r_1 = a \bmod b$ $q_1 = \lfloor a/b \rfloor$	$a = q_1 b + r_1$	$x_1 = x_{-1} - q_1 x_0 = 1$ $y_1 = y_{-1} - q_1 y_0 = -q_1$	$r_1 = ax_1 + by_1$
$r_2 = b \bmod r_1$ $q_2 = \lfloor b/r_1 \rfloor$	$b = q_2 r_1 + r_2$	$x_2 = x_0 - q_2 x_1$ $y_2 = y_0 - q_2 y_1$	$r_2 = ax_2 + by_2$
$r_3 = r_1 \bmod r_2$ $q_3 = \lfloor r_1/r_2 \rfloor$	$r_1 = q_3 r_2 + r_3$	$x_3 = x_1 - q_3 x_2$ $y_3 = y_1 - q_3 y_2$	$r_3 = ax_3 + by_3$
$\cdot$ $\cdot$ $\cdot$	$\cdot$ $\cdot$ $\cdot$	$\cdot$ $\cdot$ $\cdot$	$\cdot$ $\cdot$ $\cdot$
$r_n = r_{n-2} \bmod r_{n-1}$ $q_n = \lfloor r_{n-2}/r_{n-1} \rfloor$	$r_{n-2} = q_n r_{n-1} + r_n$	$x_n = x_{n-2} - q_n x_{n-1}$ $y_n = y_{n-2} - q_n y_{n-1}$	$r_n = ax_n + by_n$
$r_{n+1} = r_{n-1} \bmod r_n = 0$ $q_{n+1} = \lfloor r_{n-1}/r_n \rfloor$	$r_{n-1} = q_{n+1} r_n + 0$		$d = \gcd(a, b) = r_n$ $x = x_n; y = y_n$

忽略表中的 x_{-1} 和 y_{-1}

主要内容

费马定理

费马定理 Fermat's Theorem

若 p 是素数，a 是正整数且不能被 p 整除，则有 $a^{p-1} \equiv 1 (\mathrm{mod} p)$.

证明：考虑小于 p 的正整数集合 $R \triangleq \{1, \ldots, p-1\}$。用 a 乘所有元素并对 p 取模，得到 $X \triangleq \{a \bmod p, \ldots, (p-1)a \bmod p\}$。因为 p 不能整除 a，所以 X 的元素都不为 0，而且互不相等。假设 $ja \equiv ka (\mathrm{mod} p)$，其中 $1 \leq j < k \leq p-1$，因为 a 和 p 互素，因此可将 a 消去，推出 $j \equiv k (\mathrm{mod} p)$。这个等式不可能成立，因此 X 中的 $p-1$ 个元素互不相等。所以 $R = X$ 只是元素顺序不同。将两个集合中所有元素相乘并对 p 取模，得到

$$a \times 2a \times \ldots \times (p-1)a \equiv [1 \times 2 \times \ldots \times (p-1)] (\mathrm{mod} p)$$

$$a^{p-1}(p-1)! \equiv (p-1)! (\mathrm{mod} p).$$

因为 $(p-1)!$ 和 p 互素，可消去 $(p-1)!$，从而得到费马定理。 □

费马定理

举例

$a = 7, p = 19, a^{p-1} \bmod p = 7^{18} \bmod 19 = ?$

$7^2 = 49 \equiv 11 (\bmod 19)$

$7^4 = 121 \equiv 7 (\bmod 19)$

$7^8 = 49 \equiv 11 (\bmod 19)$

$7^{16} = 121 \equiv 7 (\bmod 19)$

$a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 (\bmod 19)$

费马定理等价形式

若 p 是素数且 a 是任意正整数，则 $a^p \equiv a \bmod p$。

注意未要求 a 与 p 互素

举例

$p = 5, a = 10, a^p = 10^5 \equiv 0 (\bmod 5) = a (\bmod p)$

欧拉函数 $\phi(n)$

定义：欧拉函数 Euler's Totient Function

定义欧拉函数 $\phi(n)$ 为比 n 小且与 n 互素的正整数的个数。习惯上 $\phi(1) = 1$。

n	$\phi(n)$
1	1
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4

n	$\phi(n)$
11	10
12	4
13	12
14	6
15	8
16	8
17	16
18	6
19	18
20	8

n	$\phi(n)$
21	12
22	10
23	22
24	8
25	20
26	12
27	18
28	12
29	28
30	8

欧拉函数 $\phi(n)$

欧拉函数的性质

p 和 q 是素数，$n = pq$，则 $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$。

证明：因为 p 是素数，所以 $\phi(p) = p - 1$。
为证明 $\phi(n) = \phi(p)\phi(q)$，考虑集合 $\{1, \ldots, pq-1\}$，不与 n 互素的集合为 $\{p, 2p, \ldots, (q-1)p\}$ 和 $\{q, 2q, \ldots, (p-1)q\}$。因为 p 和 q 互素，所以这两个集合不重叠：假设存在 $1 \le i \le q-1$ 和 $1 \le j \le p-1$，满足 $ip = jq$，两边模 p 得 $jq \bmod p = 0$，因为 p, q 为素数，故 $jq \bmod p \neq 0$，所以这两个集合不可能有交集。
两个集合共有 $p - 1 + q - 1$ 个整数，所以

$$\begin{aligned}\phi(n) &= (pq-1) - (p-1+q-1) \\ &= (p-1)(q-1) \\ &= \phi(q)\phi(q)\end{aligned}$$

欧拉定理

欧拉定理 Euler's Theorem

对于任意互素的 a 和 n，有 $a^{\phi(n)} \equiv 1 (\mathrm{mod}\ n)$。

证明：$\phi(n)$ 是小于 n 且与 n 互素的正整数的个数，考虑这些整数所组成的集合 $R \triangleq \{x_1, \ldots, x_{\phi(n)}\}$。将 a 与 R 中的每个元素相乘然后模 n，得到 $X \triangleq \{ax_1 \bmod n, \ldots, ax_{\phi(n)} \bmod n\}$。$X$ 是 R 的一个排列，因为

- $\gcd(a, n) = 1 \wedge \gcd(x_i, n) = 1 \Rightarrow \gcd(ax_i, n) = 1$，又因为 $\gcd(ax_i, n) = \gcd(ax_i \bmod n, n)$（欧几里得算法），所以 $\gcd(ax_i \bmod n, n) = 1$，即 X 中的每个元素与 n 互素；
- X 中没有重复元素：若 $ax_i \bmod n = ax_j \bmod n$，则 $x_i = x_j$。

所以

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

整理后得到 $a^{\phi(n)} \equiv 1 (\mathrm{mod}\ n)$。 □

欧拉定理

举例

$a = 3, n = 10, \phi(n) = 4, a^{\phi(n)} = 3^4 = 81 \equiv 1 (\mathrm{mod}\ 10) = 1 (\mathrm{mod}\ n)$

另一种表述

$a^{\phi(n)+1} \equiv a (\mathrm{mod}\ n)$

注意没有要求 a 与 n 互素

主要内容

素性测试

- 密码学中常常需要寻找大素数
- 传统的方法是用试除法，即依次除小于该数平方根的所有整数，这种方法只对较小的数有用
- 可以采用基于素数特性的统计素性测试方法
  - 其中所有的素数都满足素数特性
  - 但是有一些被称为伪素数的合数也满足素数特性
- 也可使用一种较慢的确定性素性测试方法

奇整数的表示

奇整数的表示

$n \geq 3$ 的奇整数可表示为 $n-1=2^k q$，其中 $k>0$, q 是奇数。

证明：注意到 $n-1$ 是偶数，可以用 2 去除 $n-1$，直到所得结果为奇数，此处共做了 k 次除法。 □

举例

$n=7: n-1=2\times 3$

$n=9: n-1=2^3\times 1$

$n=13: n-1=2^2\times 3$

素数的两个性质

性质一

若 p 是素数，a 是小于 p 的正整数，则 $a^2 \bmod p = 1$ 当且仅当 $a \bmod p = 1$ 或 $a \bmod p = -1 \bmod p = p - 1$。

证明：运用模运算的算术规则：$(a \bmod p)^2 \bmod p = a^2 \bmod p$。因此，若 $a \bmod p = 1$ 或 $a \bmod p = -1$，则有 $a^2 \bmod p = 1$。反之，若 $a^2 \bmod p = 1$，则 $(a \bmod p)^2 = 1$，只能是 $a \bmod p = 1$ 或 $a \bmod p = -1 \bmod p$ 之一成立。 □

注意

p 为合数时也可能成立，比如 $p = 4, a = 1$ 或 3 时，有 $a^2 \bmod p = 1$。

素数的两个性质

性质二

设 p 是大于 2 的素数，有 $p-1=2^k q$，$k>0$，q 是奇数。设 a 是整数且 $1<a<p-1$，则以下两个条件之一成立：

- $a^q \bmod p = 1$；
- 在整数 $a^q, a^{2q}, \ldots, a^{2^{k-1}q}$ 中存在一个数，和 -1 模 p 同余，即存在一个 $j, 0 \le j \le k-1$，满足 $a^{2^j q} \bmod p = -1 \bmod p = p-1$。

素数的两个性质

证明：因为 p 是素数，则由费马定理可知，$a^{p-1} \equiv 1 (\mathrm{mod} p)$。由于 $p-1=2^k q$，则 $a^{2^k q} \bmod p = 1$。观察下述数列：

$$a^q \bmod p, a^{2q} \bmod p, \ldots, a^{2^{k-1}q} \bmod p, a^{2^k q} \bmod p = 1$$

我们知道这个数列最后一个数为 1，而且每个数为前一个数的平方。最后一个数为 1，那么前一个数只能为 1 或 $p-1$；如果倒数第二个数为 1，则它前一个数只能为 1 或 $p-1$；依次类推。所以，这个数列要么全是 1，即第一个数为 1；要么数列中某个数为 $p-1$，从这个数之后数列全为 1。

因此，下面两条必有一条成立：

1. 数列的第一个数为 1；
2. 数列中某些数为 $p-1$。

□

Miller-Rabin 素性测试

- 若 n 为素数，那么数列 $a^q, a^{2q}, \ldots, a^{2^k q}$ 中；
- 要么第一个数为 1，要么数列中某个数模 n 为 $n-1$；
- 否则，n 为合数；
- 另一方面，如果上述条件满足，也不一定推出 n 为素数。

素性测试举例

例如，$n = 2047 = 23 \times 89$，则 $n - 1 = 2 \times 1023$。计算 $2^{1023} \bmod 2047 = 1$，所以虽然 $n = 2047$ 满足条件，但不是素数。

Miller-Rabin 素性测试

算法 1: Test(n)

输入: 奇整数 n
输出: n 是不是素数
找出整数 k, q，其中 $k > 0$，q 是奇数，使 $n - 1 = 2^k q$
随机选取整数 $a, 1 < a < n - 1$
if $a^q \bmod n = 1$ **then** 返回“可能是素数”
for $j = 0$ to $k - 1$ **do**
  if $a^{2^j q} \bmod n = n - 1$ **then** 返回“可能是素数”
返回“合数”

重复使用 Miller-Rabin 算法

- 如果算法返回 “合数”，则这个数必为合数；不然有可能为素数；
- 据相关文献，给定一个非素奇数 n 和一个随机整数 $a, 1 < a < n-1$，程序 TEST 返回不确定的概率小于 $1/4$（即不能确定 n 是不是素数）；
- 因此，如果选择 t 个不同 a 进行测试，则它们都能通过测试（返回不确定）的概率小于 $(1/4)^t$；
- 这为决定一个奇整数 n 是素数且具有合理的可信度奠定了基础：对随机选取的 a, 重复调用 TEST(n), 如果某时刻 TEST 返回 “合数”, 则 n 一定不是素数；若 TEST 连续 t 次返回 “不确定”, 当 t 足够大时, 我们可以相信 n 是素数。

Miller-Rabin 素性测试举例

举例：考虑素数 $n = 29$

- $n - 1 = 28 = 2^2 \times 7 = 2^k q$
- 选取 $a = 10$
  - $a^q \bmod n = 17$，它既不是 1 也不是 $n - 1 = 28$
  - $a^{2q} \bmod n = 28$；返回 “有可能是素数”
- 选取 $a = 2$
  - $a^q \bmod n = 12$
  - $a^{2q} \bmod n = 28$；返回 “有可能是素数”

举例：考虑合数 $n = 13 \times 17 = 221$

- $n - 1 = 220 = 2^2 \times 55 = 2^k q$
- 选取 $a = 5$
  - $a^q \bmod n = 112$
  - $a^{2q} \bmod n = 168$；返回 “合数”

Miller-Rabin 素性测试举例

举例：考虑素数 $n = 29$

- $n - 1 = 28 = 2^2 \times 7 = 2^k q$
- 选取 $a = 10$
  - $a^q \bmod n = 17$，它既不是 1 也不是 $n - 1 = 28$
  - $a^{2q} \bmod n = 28$；返回 “有可能是素数”
- 选取 $a = 2$
  - $a^q \bmod n = 12$
  - $a^{2q} \bmod n = 28$；返回 “有可能是素数”

举例：考虑合数 $n = 13 \times 17 = 221$

- $n - 1 = 220 = 2^2 \times 55 = 2^k q$
- 选取 $a = 5$
  - $a^q \bmod n = 112$
  - $a^{2q} \bmod n = 168$；返回 “合数”

确定性素性判定方法 AKS

- 2002 年以前，没有高效的方法证明一个大数的素性，包括 Miller-Rabin 算法在内，所有在用算法给出的都是概率性结果。
- 2002 年 Agrawal, Kayal 和 Saxena 给出了一个相对简单的确定性算法 AKS，可以有效判定一个大数是否为素数，但是看上去没有 Miller-Rabin 算法快，因此没有代替古老的概率算法。

素数的分布

- 由数论中的素数定理可知，n 附近的素数分布情况为平均每 $\ln n$ 个整数中有一个素数。平均而言，在找到一个素数之前必须测试约 $\ln n$ 个整数
- 因为偶数肯定不是素数，因此需要测试的整数个数为 $0.5 \ln n$。例如，若要找 2200 左右的素数，则约需要 $0.5 \ln 2200 = 69$ 次测试。
- 这只是个平均值，在数轴上的某些位置，素数非常密集，而在其他有些位置，素数非常稀疏。
  - 两个相邻的奇数 $1,000,000,000,061$ 和 $1,000,000,000,063$ 都是素数
  - 而 $1001! + 2, 1001! + 3, \ldots, 1001! + 1000, 1001! + 1001$ 这 1000 个连续的整数都是合数

主要内容

单向函数与单向陷井门函数

定义：单向函数，One-way Function

函数 f 若满足下列条件, 则称 f 为单向函数：

1. 对于所有属于 f 之域的任一 x，容易计算 $y = f(x)$
2. 对于几乎所有属于 f 之域的任一 y，求得 x，使 $y = f(x)$，在计算上不可行

定义：单向陷井门函数，One-way Trapdoor Function

函数 F 若满足下列两条件, 则称 F 为单向陷井门函数：

1. 对于所有属于 F 之域的任一 x，容易计算 $y = F(x)$；
2. 对于几乎所有属于 F 之域的任一 y，除非获得暗门信息 (trapdoor)，否则求出 x，使得 $x = F^{-1}(y)$ 在计算上不可行，F^{-1} 为 F 之逆函数；如有额外信息（暗门），则容易求出 $x = F^{-1}(y)$

单向函数举例：离散对数问题（Discrete Logarithm Problem, DLP）

- p 为素数，整数 g 满足 $1 < g < p - 1$。
- 若给定整数 x，求 $y = g^x \bmod p$，最多需要 $\lfloor log_2 x \rfloor + w(x) - 1$ 次乘法，$w(x)$ 为 x 中所有 1 的个数。
  - 如 $x = 15$, 即 $x = (1111)_2, w(x) = 4$, 则 $g^{15} = (g^2 g)^2 g)^2 g \bmod p$, 只需要 $3 + 4 - 1 = 6$ 次乘法。
- 但是若给定 p, g 及 y, 求 x, 则为 DLP 问题, 最快方法需要 $L(p) = \exp\{(\ln p)^{1/3} (\ln \ln p)^{2/3}\}$ 次运算。
  - 当 $p = 512$ 位时, $L(p)$ 约为 $2^{256} \approx 10^{77}$, 计算上不可行, 因为 $2^{100} \approx 10^{30}$, 计算要 10^{16} 年。

单向函数举例：因数分解问题（Factorization Problem, FAC）

- 给定大素数 p 和 q，求 $n = p \times q$，只要一次乘法
- 给定 n，求 p 和 q，即为因数分解问题，最快方法需要 $e^{c\sqrt{\ln n \ln\ln n}}$ 次运算，其中 c 为大于 1 的正整数。若 $p \approx n$，解离散对数比因数分解难。

单向函数举例：背包问题（Knapsack Problem）

- 给定有限个自然数序列集合 $B = (b_1, b_2, \ldots, b_n)$ 及二进制序列 $x = (x_1, x_2, \ldots, x_n), x_i \in \{0, 1\}$, 求 $S = \sum_i x_i b_i$ 最多只需 $n - 1$ 次加法；但若给定 B 和 S, 求 x 则非常困难。
- 穷举时有 2^n 种可能, 当 n 很大时为计算上不可行。
- Garey 和 Johnson 证明, 背包问题是 NP 难问题。

单向函数及其交换性

单向函数本身对近代密码学领域用处不大，但若具有交换性，则作用大。

定义：交换性 Commutative Property

令 Z 为一集合，F 为将 Z 映射到 Z 本身的函数集合。令 $z \in Z$，$F_x(z)$ 表示此函数集合之第 x 函数, 若 $F_x(F_y(z)) = F_y(F_x(z))$，则称此函数集合具有交换性。

例如 $D(E(m)) = E(D(m))$

指数函数 Exponential Function

定义：指数函数

令 G 为有限乘法群，$g \in G$，则对于所有整数 x，$E_x(g) = g^x \bmod p \in G$ 称为指数函数。

通常，令 $G = \{1, 2, \ldots, p-1\}$，$p$ 为素数，则 $E_x(g) = g^x \bmod p$ 为指数函数。

指数函数的特性：周期性，Periodicity

- 令序列 $\langle E_x(g)\rangle = \{g^0 \bmod p, g^1 \bmod p, g^2 \bmod p, \ldots\}$ 为 g 所产生之序列。
- 因为 G 是有限群, $E_x(g)$ 不可能不重复, 故 $E_x(g)$ 产生之序列为周期序列。
- 当存在最小正整数 T, 使得 $E_T(g) = g^T \equiv 1 (\bmod p)$ 时, 称 T 为 g 在 G 中的阶或序（order)、周期。
- 根据费马定理，对于所有 g, T 必定整除 $p - 1$.

举例

$p = 11, g = 2$

x	0	1	2	3	4	5	6	7	8	9	10
$g^x \bmod p$	1	2	4	8	5	10	9	7	3	6	1

所以 $T = 10$

指数函数的特性：周期性，Periodicity

Table 2.7 Powers of Integers, Modulo 19

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

指数函数的特性：本原元（素根、原根），Primitive Element (Root)

定义：本原元

若 $g \in G$ 的阶为 $T = p - 1$，则称 g 为模 p 的本原元。

- 当 g 为 $\mathrm{mod} p$ 的本原元时，由 g 产生的序列 $\langle E_x(g) \rangle$ 具有最大周期（安全性较高）。
- 对于所有素数 p，其本原元必定存在。
- 当 g 为模 p 的本原元且 a 与 $p - 1$ 互素时，即 $\gcd(a, p - 1) = 1$，则 $g^a \bmod p$ 亦必为模 p 之本原元。
- 模 p 的本原元素个数为 $\phi(p - 1)$。

指数函数的特性：本原元

本原元举例

- $p = 11, g = 2, \phi(p-1) = \phi(10) = 4$, 即 $1, 3, 7, 9$ 与 $p-1$ 互素。
- 若 $g = 2$ 为模 p 之本原元素, 则 $2^1 \bmod 11 = 2$，$2^3 \bmod 11 = 8$，$2^7 \bmod 11 = 7$，$2^9 \bmod 11 = 6$，均为模 11 之本原元素。
- 找到一个本原元素后可以容易找到所有本原元素, 问题是如何找到第一个本原元素。

指数函数的特性

（注意：以下运算中省略了 $\bmod p$ 操作）

3. 交换性

- 因为

$$E_x(E_y(g)) = E_x(g^y) = (g^y)^x = g^{yx}$$
$$E_y(E_x(g)) = E_y(g^x) = (g^x)^y = g^{xy}$$

- 所以 $E_x(E_y(g)) = E_y(E_x(g))$

4. 非对称性（Asymmetric Property）

- $E_x(-g) = (-g)^x = (-1)^x g^x = (-1)^x E_x(g)$
- 若 x 为偶，则 $E_x(-g) = E_x(g)$
- 若 x 为奇，则 $E_x(-g) = -E_x(g)$

5. 乘法性（Asymmetric Property）

- $E_x(g_1)E_x(g_2) = g_1^x g_2^x = (g_1 g_2)^x = E_x(g_1 g_2)$

指数函数的特性：乘法逆元，Multiplicative Inverse

- 若 T 为 g 之序，则对于所有 x，$0 \leq x < T$，$E_x(g^{-1}) = E_{T-x}(g)$
- g^{-1} 为 g 的乘法逆元素.
- 因为：$E_x(g^{-1}) = g^{-x} = 1 \cdot g^{-x} = g^T \cdot g^{-x} = g^{T-x} = E_{T-x}(g)$
- 这是一种求乘法逆元素的方法:
  - 欲求 g^{-1} 时，由于 $g^{T-1} = g^{-1}$（这里 $x = 1$）
  - 因为 T 整除 $p - 1$
  - 所以 $g^{-1} = g^{T-1} = g^{p-1-1} = g^{p-2} (\mathrm{mod} p)$
  - $g \cdot g^{-1} \bmod p = 1$，这是因为 $g^x g^{T-x} \bmod p = g^T \bmod p = 1$

指数函数的特性

7. 安全性

- 给定 $g \in G$ 及 $y \in \langle E_x(g) \rangle$，求 x 使得 $y = E_x(g) = g^x \bmod p$ 为 DLP 问题。

8. 可逆性

- 若 T 为 $g \in G$ 之序，x^{-1} 为 x 在模 T 时的乘法逆元素，即 $xx^{-1} \equiv 1 (\bmod T)$
- 则 $E_x(E_{x^{-1}}(g)) = E_{x^{-1}}(E_x(g)) = g^{xx^{-1}} \bmod p = g \bmod p$
- 因为 $E_x(g)$ 有交换性, 所以 $E_x(E_{x^{-1}}(g)) = E_{x^{-1}}(E_x(g))$。
- 另一种证明方法：
  - 因为 $x^{-1}x = 1 \bmod T = kT + 1$
  - 所以 $E_x(E_{x^{-1}}(g)) = g^{xx^{-1}} = g^{kT+1} = (g^T)^k \cdot g = 1^k g = g \bmod p$
  - 这实际上是利用费马定理对 RSA 算法正确性的证明

快速指数运算

- 问题：当 x 是一个大整数时，如何快速计算 g^x?
- 当 x 为 n 位时，即 $x = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$
- $E_x(g) = g^x = ((g^{x_{n-1}})^2 \cdot g^{x_{n-2}})^2 \cdot g^{x_{n-3}} \ldots)^2 \cdot g^{x_0}$
- 此算法共需要 $n-1$ 次平方及 $w(x)-1$ 次乘法，其中 $w(x)$ 为 x 的二进制表示中 1 的个数。
- 平均而言 $w(x) = n/2$，x 在二进制表示时有 $n/2$ 个 0 及 $n/2$ 个 1。
- 因此，当 x 为 n 位时，平均需要 $1.5n-2$ 个乘法（平方算一次乘）。

快速指数运算

```
long FastExp(long a,long n){
  long base = a;
  long res = 1;
  while(n!=0){
    if((n&1) == 1) res *= base;
    base *= base;
    n >>= 1;
  }
  return res;
}
```

主要内容

计算乘法逆元

- 问题：$ax \bmod n = 1$，$x = a^{-1} = ?$
- 根据欧拉定理：若 $\gcd(a, n) = 1$，则 $a^{\phi(n)} \bmod n = 1$.
- 因此，$x = a^{\phi(n)-1} \bmod n$.
- 如果 $\phi(n)$ 已知, 则 a 的逆元可以用快速指数运算法求得。
  - 特例：如果 n 是素数, 则 $\phi(n) = n - 1$，所以 $a^{-1} = a^{n-1-1} \bmod n = a^{n-2} \bmod n$.
- 如果 $\phi(n)$ 未知, 可以用扩展 Euclid 算法来求逆。

在 $GF(2^n)$ 中求逆元

- 因为除了 0，$GF(2^n)$ 中每个元素都与素多项式 $p(x)$ 互素，所以 $\phi(p(x)) = 2^n - 1$
- 所以 $a^{-1} = a^{\phi(p(x))-1} \bmod p(x) = a^{2^n-2} \bmod p(x)$

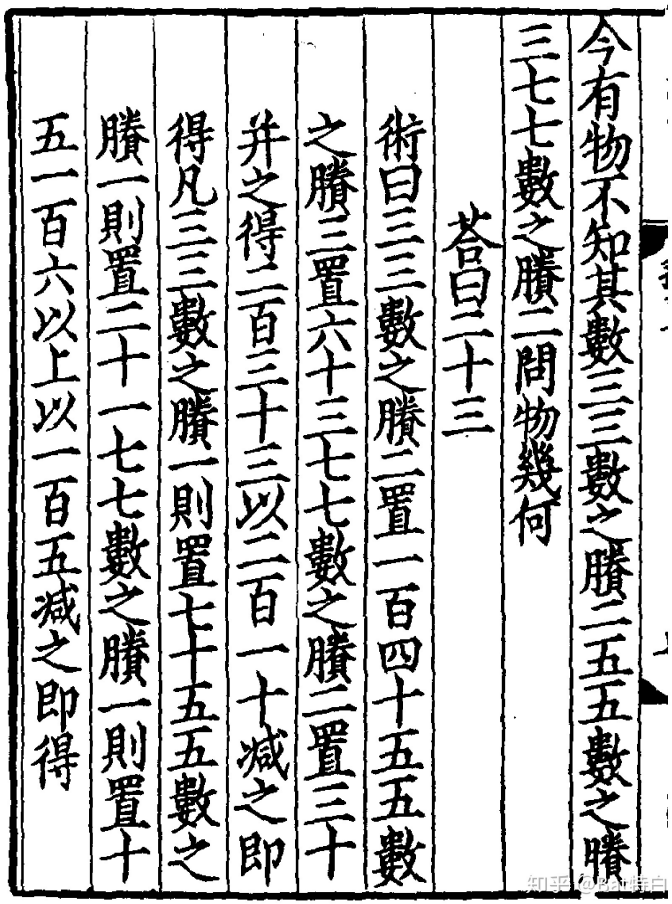
举例

在 $GF(2^3)$ 中，$a = 100$，$p(x) = 1011$ $a^{-1} = a^{\phi(x)-1} \bmod p(x) = a^{2^3-2} \bmod p(x) = 100^6 \bmod 1011 = 111$

主要内容

中国余数定理 Chinese Remainder Theorem (CRT)

- 也称为“孙子定理”。
- 一元线性同余问题最早可见于中国南北朝时期（公元 5 世纪）的数学著作《孙子算经》中的“物不知其数”问题。
- 中国余数定理说明某一范围内的整数可通过它对两两互素的整数取模所得的余数来重构。


今有物不知其數三三數之賸二五五數之賸二七七數之賸二問物幾何
答曰二十三
術曰三三數之賸二置一百四十五五數之賸三置六十三七七數之賸二置三十并之得二百三十三以二百一十減之即得凡三三數之賸一則置七十五五數之賸一則置二十一七七數之賸一則置十五一百六以上以一百五減之即得

举例：如何由余数重构整数

- $Z_{10} = \{0, 1, \ldots, 9\}$ 中的 10 个整数可通过它们对 2 和 5（10 的素因子）取模所得的两个余数来重构。
- 假设数 x 的余数 $r_2 = 0$ 且 $r_5 = 3$，即 $x \bmod 2 = 0$，$x \bmod 5 = 3$，
- 则 x 是 Z_{10} 中的偶数且被 5 除余 3，唯一解 $x = 8$。

CRT 的几种表述形式

令 $n_1, \ldots, n_k$ 两两互素，$n = \prod_i n_i$，则

- Z_n 中的任一整数 $a \in Z_n$ 都对应一个 k 元组 $(x_1, \ldots, x_k)$，其中 $x_i = a \bmod n_i, i = 1, \ldots, k$。
- 一元线性同余方程组

$$\begin{cases} x \equiv x_i \pmod{n_i} \\ \cdots \\ x \equiv x_k \pmod{n_k} \end{cases}$$

在 Z_n 中有一个公共解 x。

CRT 的几种表述形式

令 $n_1, \ldots, n_k$ 两两互素，$n = \prod_i n_i$，则

- Z_n 中的任一整数 $a \in Z_n$ 都对应一个 k 元组 $(x_1, \ldots, x_k)$，其中 $x_i = a \bmod n_i, i = 1, \ldots, k$。

- 一元线性同余方程组

$$\begin{cases} x \equiv x_i \pmod{n_i} \\ \ldots \\ x \equiv x_k \pmod{n_k} \end{cases}$$

在 Z_n 中有一个公共解 x。

证明

- $\forall i$，因为 $\gcd(n_i, n/n_i) = 1$，所以存在 y_i 使得 $(n/n_i)y_i \bmod n_i = 1$（即因为 n/n_i 与 n_i 互素，所以 n/n_i 存在模 n_i 的逆元 y_i）。
- $\forall i \neq j$，因为 n_j 是 n/n_i 的一个因子，则 $(n/n_i)y_i \bmod n_j = 0$。
- 令

$$x \triangleq \left[\sum_{i=1}^{k} (n/n_i)y_i x_i\right] \bmod n.$$

因为

$$x \bmod n_i = (n/n_i)y_i x_i \bmod n_i = x_i$$

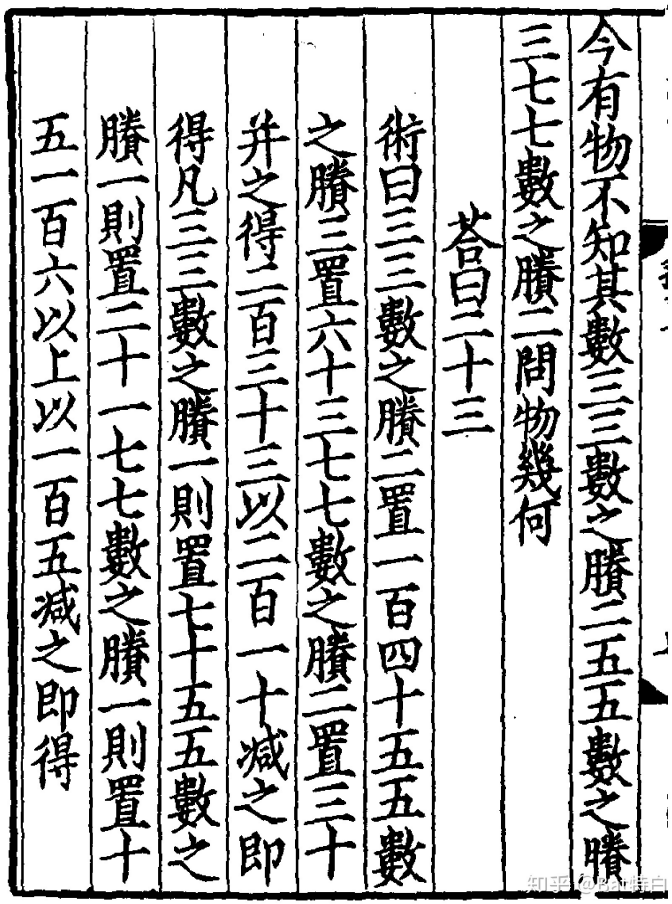
所以 x 是 $x \bmod n_i = x_i, i = 1, \ldots, k$ 的公共解。

“物不知其数”问题求解

$x \bmod 3 = 2$
$x \bmod 5 = 3$
$x \bmod 7 = 2$

$n_1 = 3, n_2 = 5, n_3 = 7$
$x_1 = 2, x_2 = 3, x_3 = 2$
$n = 3 \times 5 \times 7 = 105$

(1) 求 y_i，使得 $(n/n_i)y_i \bmod n_i = 1$
得：$y_1 = 2, y_2 = 1, y_3 = 1$
(2) $x = \sum_i (n/n_i)y_i x_i \bmod n$
$= (70 \times 3 + 21 \times 5 + 15 \times 7) \bmod 105$
$= 23$


今有物不知其數三三數之賸二五五數之賸三七七數之賸二問物幾何
答曰二十三
術曰三三數之賸二置一百四十五五數之賸三置六十三七七數之賸二置三十并之得二百三十三以二百一十減之即得凡三三數之賸一則置七十五五五數之賸一則置二十一七七數之賸一則置十五一百六以上以一百五減之即得

“物不知其数”问题求解

《孙子歌诀》

三人同行七十希，
五树梅花廿一支，
七子团圆正半月，
除百零五便得知。

明朝数学家程大位《算法统宗》