



第 4 章：公钥密码学

赵俊舟

junzhou.zhao@xjtu.edu.cn

2025 年 3 月 28 日

目录

- 1 数论基础
- 2 基本概念与 RSA 算法
- 3 Diffie-Hellman 密钥交换协议
- 4 椭圆曲线密码

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- 素性测试
- 单向函数与指数函数
- 计算乘法逆元
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- 素性测试
- 单向函数与指数函数
- 计算乘法逆元
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

数论简介

- 数论主要研究整数集合 \mathbb{Z} 的性质，尤其关注正整数集合 \mathbb{Z}^+

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\} \quad \mathbb{Z}^+ = \{1, 2, 3, \dots\}$$

- 按数的可分性，一个正整数总能分为以下三类：

- 单位元：1
 - 素数：2, 3, 5, 7, 11, 13, 17, 19, ...
 - 合数：4, 6, 8, 9, 10, 12, 14, 15, ...
- 一个正整数 $p > 1$ 是素数当且仅当它只有因子 1 和 p 。
 - 素数是数论的核心，因为所有整数 $n > 1$ 都可以进行素因子分解：

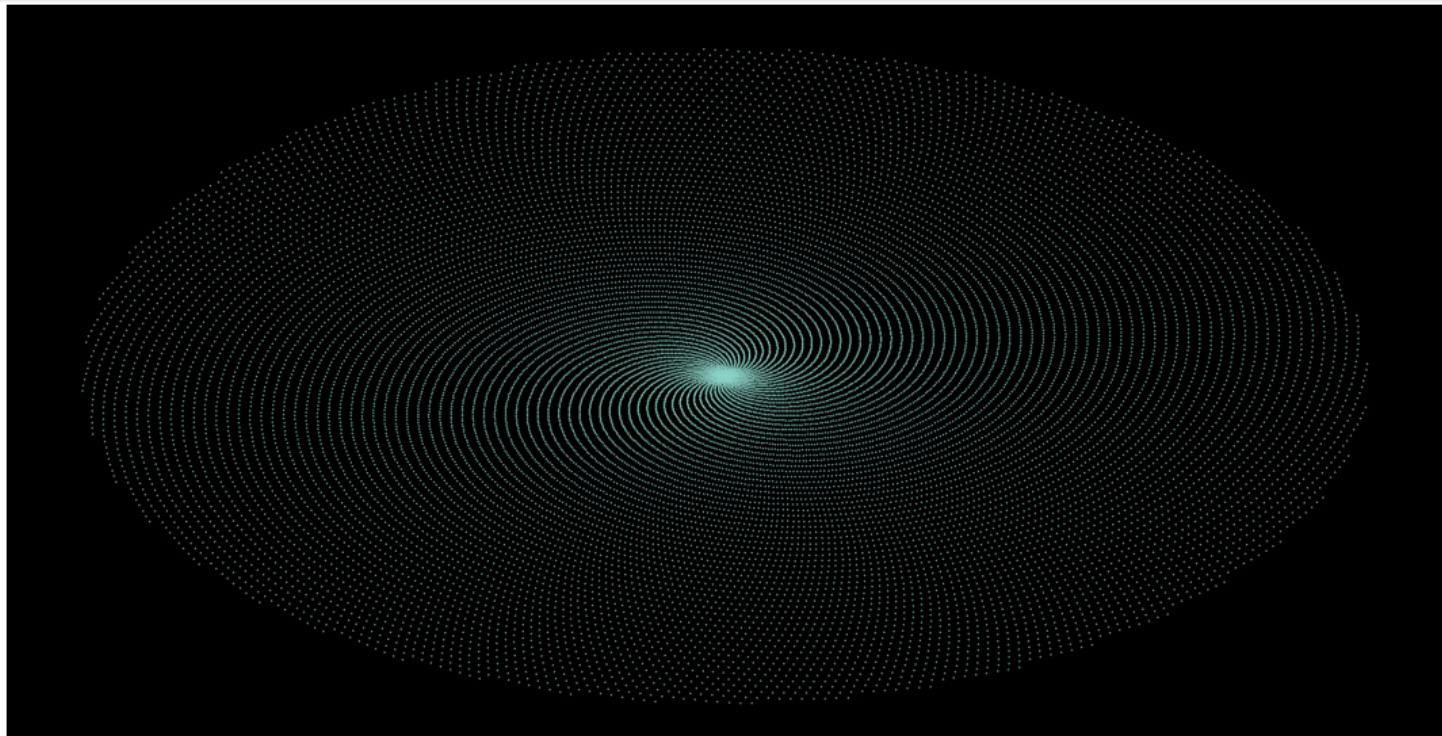
$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$$

其中 $p_1 < p_2 < \cdots < p_k$ 为素数， $\alpha_1, \alpha_2, \dots, \alpha_k$ 为正整数。

关于素数的研究问题

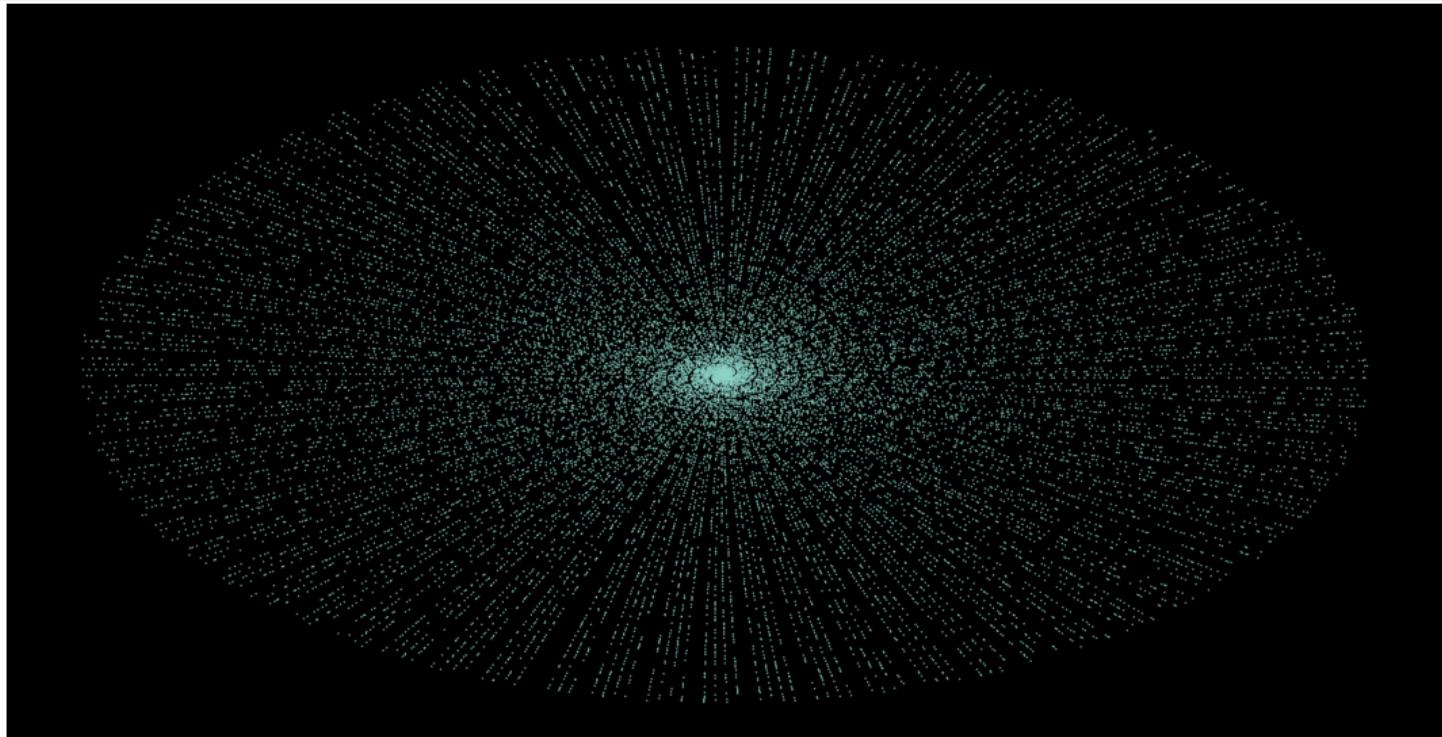
- 数论已经被研究了 2000 多年，关于素数仍有许多问题悬而未决，例如
 - Q1: 素数的分布情况**
做出关键贡献的学者：欧几里得 (300BC)、黎曼 (1859)、阿达马和普桑 (1896) 等
 - Q2: 孪生素数的分布情况**
做出关键贡献的学者：哥德巴赫 (1742)、陈景润 (1966)、张益唐 (2013) 等
 - Q3: 等差素数列的分布情况**
做出关键贡献的学者：陶哲轩 (2007) 等
- 数论中的问题通常很容易表述，但这些问题往往很难解决。

Q1: 素数的分布 (前 2 万个整数的分布)



$$x \rightarrow (x \cos(x), x \sin(x))$$

Q1: 素数的分布 (前 2 万个素数的分布情况)



$$x \rightarrow (x \cos(x), x \sin(x))$$

Q1: 素数的分布

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																			

Q1: 素数的分布

- 欧几里得在其著作《几何原本》中证明有无穷多个素数。
- 最小的素数是 2, 目前发现的最大素数是 $2^{136,279,841} - 1$ (发现于 2024 年 10 月 21 日), 共 41,024,320 位数, 比上一个发现的最大素数 (2018 年) 多 1,600 位。
- 用 $\pi(x)$ 表示不超过 x 的素数个数, 欧几里得定理其实说明
$$\pi(x) \rightarrow \infty \quad \text{as} \quad x \rightarrow \infty$$
- 更准确的结论是素数定理, 由阿达马 (1896) 等人证明

$$\pi(x) \sim \frac{x}{\ln x}$$

- 可以近似认为, 在区间 $[1, x]$ 碰到一个素数的概率为 $1/\ln x$; 或者说, 在 x 附近, 每 $\ln x$ 个整数中有一个素数。

Q1: 素数的分布

- 如果黎曼猜想为真，那么素数定理可以进一步精确为

$$\pi(x) = \int_2^x \frac{dt}{\ln t} + O(xe^{-c\sqrt{\ln x}})$$

- 黎曼猜想是复分析中的一个著名猜想：复平面上 ζ 函数

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}, \quad s = \sigma + it, \sigma, t \in \mathbb{R}$$

满足 $0 < \operatorname{Re}(s) < 1$ 的所有非平凡零点都位于 $\operatorname{Re}(s) = 1/2$ 上，即满足 $\zeta(\rho) = 0$ 的点 ρ 具有形式 $\rho = 1/2 + it$ 。

- 黎曼猜想是克雷数学研究所于 2000 年提出的 7 个千禧年大奖难题之一，每个难题奖金 100 万美元。

Q2: 孪生素数的分布

- 孪生素数指相差为 2 的素数对，例如 $(3, 5), (5, 7), (11, 13)$ 等。
目前发现的最大孪生素数为（发现于 2016 年）：

$$2,996,863,034,895 \times 2^{1,290,000} \pm 1$$

- 用 $\pi_2(x)$ 表示不超过 x 的孪生素数数量，孪生素数猜想说明

$$\pi_2(x) \rightarrow \infty \quad \text{as} \quad x \rightarrow \infty$$

- 如果素数分布服从独立同分布，那么

$$\pi_2(x) \sim \frac{x}{(\ln x)^2}$$

- 素数分布显然不独立，哈代和利特尔伍德猜想：

$$\pi_2(x) = 2 \prod_{p \geq 3} \frac{p(p-2)}{(p-1)^2} \int_2^x \frac{dt}{(\ln t)^2} \approx 1.320323632 \int_2^x \frac{dt}{(\ln t)^2}$$

Q2: 孪生素数的分布

- 陈景润利用筛法证明：有无穷多个整数对 $(p, p + 2)$ 其中 p 为素数， $p + 2$ 是不超过 2 个素数的乘积（1966–1973）。
- 张益唐证明：间距小于 7000 万的素数对有无穷多个（2013）。
- 仅仅过了几个月，素数对之差被缩小为 246。
- 差是 2 的素数对为孪生素数对，差是 4 的素数对为表亲素数对，差是 6 的素数对为性感素数对……。
- 波里尼亞克猜想**（也称为**弱孪生素数猜想**，1849）：存在无穷多个素数对 $(p, p + 2k)$, $k = 1, 2, 3, \dots$ 。

Q3: 等差素数列的分布

- 等差素数列是如下形式的素数数列

$$p, p + d, p + 2d, \dots, p + kd$$

其中 p 是首项, d 是公差, $p + kd$ 是尾项。例如 $(3, 5, 7)$, $(5, 11, 17, 23, 29)$ 。

- 目前发现的最长等差素数列为 (记为 AP27, 2009)
 $224, 584, 605, 939, 537, 911 + 81292139 \cdot 23\# \cdot k \quad k = 0, \dots, 26$
其中 $23\#$ 是不超过 23 的素数的乘积。
- 格林-陶定理 (2004): 存在任意长的等差素数列。
- 陶哲轩等人于 2006 年获菲尔兹奖, 等同于数学诺贝尔奖。
- 目前仍不清楚怎样去发现任意长等差素数列, 也不清楚等差连续素数列的存在情况。

关于素数的参考资料

- Great Internet Mersenne Prime Search:
<https://www.mersenne.org>
- The largest known simultaneous primes:
<http://primerecords.dk/simulprime.htm>
- 目前发现的最长等差素数列：
<http://primerecords.dk/aprecords.htm>
- 目前发现的最长等差连续素数列：
<http://primerecords.dk/cpap.htm>

目录

1 数论基础

- 数论简介
- **费马定理和欧拉定理**
- 素性测试
- 单向函数与指数函数
- 计算乘法逆元
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

费马定理

定理 (费马定理 Fermat's Little Theorem)

若 p 是素数, a 是正整数且不能被 p 整除, 则有

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

证明.

- 考虑小于 p 的正整数集合 $R \triangleq \{1, \dots, p-1\}$ 。用 a 乘所有元素并对 p 取模, 得到 $X \triangleq \{a \bmod p, \dots, a(p-1) \bmod p\}$ 。
- $\gcd(a, p) = 1 \Rightarrow X$ 的元素都不为 0 且互不相等 $\Rightarrow R = X$ 。
- 将两个集合中所有元素相乘并对 p 取模, 得到

$$a \times 2a \times \dots \times a(p-1) \equiv 1 \times 2 \times \dots \times (p-1) \pmod{p}$$

$$a^{p-1}(p-1)! \equiv (p-1)! \pmod{p}$$

因为 $(p-1)!$ 和 p 互素, 可以消去, 从而得到费马定理。 □

费马定理

例 ($a = 7, p = 19$)

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

$$7^2 = 49 \equiv 11 \pmod{19}, 7^4 = 121 \equiv 7 \pmod{19}$$

$$7^8 = 49 \equiv 11 \pmod{19}, 7^{16} = 121 \equiv 7 \pmod{19}$$

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

推论 (费马定理等价形式)

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

注意，这里不要求 a 与 p 互素。

例 ($a = 10, p = 5$)

$$a^p = 10^5 \equiv 0 \pmod{5} = a \pmod{p}$$

欧拉函数 (Euler's Totient Function)

定义 (欧拉函数)

欧拉函数, 记作 $\phi(n)$, 是比 n 小且与 n 互素的正整数的个数。

习惯上约定 $\phi(1) = 1$ 。

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

性质

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

证明.

- 考虑集合 $\{1, \dots, pq - 1\}$, 其中不与 pq 互素的数的集合为 $\{p, 2p, \dots, (q - 1)p\}$ 和 $\{q, 2q, \dots, (p - 1)q\}$ 。
- 因为 p 和 q 互素, 所以这两个集合无交集: 假设存在 $1 \leq i \leq q - 1$ 和 $1 \leq j \leq p - 1$, 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

$$\phi(n) = (pq - 1) - (p - 1 + q - 1) = (p - 1)(q - 1) = \phi(p)\phi(q)$$



欧拉函数

性质

p 是素数, 则 $\phi(p^k) = p^{k-1}(p - 1)$ 。

性质

正整数 n 的素因子分解为 $n = p_1^{k_1} p_2^{k_2} \cdots p_t^{k_t}$, 则

$$\phi(n) = \prod_{i=1}^t p_i^{k_i-1} (p_i - 1)$$

- 对一个正整数 n 进行素因子分解是很困难的事情, 因此目前尚不存在通用的计算 $\phi(n)$ 的高效算法。
- 数论中可以证明, 计算一个正整数 n 的欧拉函数 $\phi(n)$ 等同于对 n 进行素因子分解。

欧拉定理 (Euler's Theorem)

定理 (欧拉定理)

对于任意互素的正整数 a 和 n , 有 $a^{\phi(n)} \equiv 1 \pmod{n}$ 。

证明.

- 令小于 n 且与 n 互素的正整数构成集合 $R \triangleq \{x_1, \dots, x_{\phi(n)}\}$, 然后将 a 与 R 中的每个元素相乘然后模 n , 得到集合 $X \triangleq \{ax_1 \bmod n, \dots, ax_{\phi(n)} \bmod n\}$ 。
- 由于

- $\gcd(a, n) = 1 \wedge \gcd(x_i, n) = 1 \Rightarrow \gcd(ax_i, n) = 1 \Rightarrow \gcd(ax_i \bmod n, n) = 1$, 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, X 其实是 R 的一个排列, $X = R$ 。

欧拉定理 (Euler's Theorem)

- 把两个集合中的元素乘起来

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

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

$$a^{\phi(n)} \prod_{i=1}^{\phi(n)} x_i \equiv \prod_{i=1}^{\phi(n)} x_i \pmod{n}$$

因为 $\gcd(\prod_{i=1}^{\phi(n)} x_i, n) = 1$ ，所以利用模运算的性质得到

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

从而得到欧拉定理。



欧拉定理 (Euler's Theorem)

- 费马定理是欧拉定理当 n 为素数时的一个特殊情况。
- 例如, $a = 3, n = 10, \phi(n) = 4, a^{\phi(n)} = 3^4 \equiv 1 \pmod{10}$

推论

n 是素数, 对任意正整数 a , 有 $a^{\phi(n)+1} \equiv a \pmod{n}$ 。

注意

上述推论要求 n 是素数, 如果 n 不为素数, 考虑下例

- $a = 2, n = 4$, 此时 $\phi(n) = 2$ 。
- $2^3 \bmod 4 = 8 \bmod 4 = 0$, 而 $2 \bmod 4 = 2$, 两者不相等。

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- **素性测试**
- 单向函数与指数函数
- 计算乘法逆元
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

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。 □

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 (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 $a \bmod p = p - 1$ 。

证明.

- \Rightarrow : 由 $a^2 \bmod p = 1$ 知 $p|(a^2 - 1)$ 即 $p|(a + 1)(a - 1)$ 。由于 p 是素数, 故只能是 $p|(a + 1)$ 或 $p|(a - 1)$, 得 $a \bmod p = 1$ 或 $a \bmod p = -1 \bmod p = p - 1$ 。
- \Leftarrow : 当 $a \bmod p = 1$ 或 $a \bmod p = p - 1$ 时, 有 $p|(a - 1)$ 或 $p|(a + 1)$, 所以 $p|(a + 1)(a - 1)$, 即 $p|(a^2 - 1)$, 从而得到 $a^2 \bmod p = 1$ 。



某些合数也可能成立。 $p = 4$, 当 $a = 1$ 或 3 时, $a^2 \bmod p = 1$

素数的两个性质

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 是小于 p 的整数, 则以下两个结论必然有一个成立:

- $a^q \bmod p = 1$ 。
- 在整数 $a^q \bmod p, a^{2q} \bmod p, \dots, a^{2^{k-1}q} \bmod p$ 中存在一个数为 $p - 1$ 。即存在 $0 \leq j \leq k - 1$, 满足 $a^{2^j q} \bmod p = p - 1$ 。

例

例如 $p = 29$ 为素数, $29 - 1 = 2^2 \times 7$ 。取 $a = 2$, 则 $a^q \bmod p = 2^7 \bmod 29 = 12$, $a^{2q} \bmod p = 2^{14} \bmod 29 = 28$, 满足第二个结论, 故该性质对素数 29 成立。

素数的两个性质

证明.

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

$$a^q \pmod{p}, a^{2q} \pmod{p}, \dots, a^{2^{k-1}q} \pmod{p}, a^{2^k q} \pmod{p} = 1$$

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 $p-1$, 从这个数之后全为 1。

从而证明了性质二。 □

Miller-Rabin 素性测试

Miller-Rabin 素性测试

- 若 n 为素数, $n - 1 = 2^k q$, $a \in \{1, \dots, n - 1\}$, 那么数列
$$a^q \bmod n, a^{2q} \bmod n, \dots, a^{2^{k-1}q} \bmod n$$
要么第一个数为 1, 要么数列中某个数为 $n - 1$ 。
- 如果不满足上一条, 那么 n 必为合数。
- 注意, 如果上述条件满足, 也不一定推出 n 一定为素数。

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: PrimeTest(n)

输入: 奇整数 n

输出: n 是不是素数

- 1 找出整数 k, q , 其中 $k > 0$, q 是奇数, 使 $n - 1 = 2^k q$ 。
- 2 随机选取整数 $a \in \{1, \dots, n - 1\}$ 。
- 3 **if** $a^q \bmod n = 1$ **then**
- 4 **return** 可能是素数。
- 5 **for** $j = 0$ to $k - 1$ **do**
- 6 **if** $a^{2^j q} \bmod n = n - 1$ **then**
- 7 **return** 可能是素数。
- 8 **return** 是合数。

重复使用 Miller-Rabin 算法

- 如果返回“合数”，则这个数必为合数。否则可能为素数。
- 有结论：给定一个非素奇数 n 和一个随机整数 $a, 1 < a < n - 1$ ，程序 PrimeTest 误报的概率小于 $1/4$ （即当程序返回“ n 可能是素数”时，误报的概率小于 $1/4$ ）。
- 因此，如果选择 t 个不同 a 进行测试，则它们都能通过测试并产生误报的概率小于 $(1/4)^t$ 。
- 对随机选取的 a ，重复调用 $\text{PrimeTest}(n)$ ，如果某时刻 PrimeTest 返回“合数”，则 n 一定不是素数。
- 若 PrimeTest 连续 t 次返回“可能是素数”，当 t 足够大时，可以相信 n 是素数。

Miller-Rabin 素性测试举例

例 (考虑素数 $n = 29$)

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

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。
- 偶数肯定不是素数, 因此需要测试 $0.5 \ln n$ 个整数。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 $1001! + 2, 1001! + 3, \dots, 1001! + 1000, 1001! + 1001$ 这 1000 个连续的整数都是合数。

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- 素性测试
- **单向函数与指数函数**
- 计算乘法逆元
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

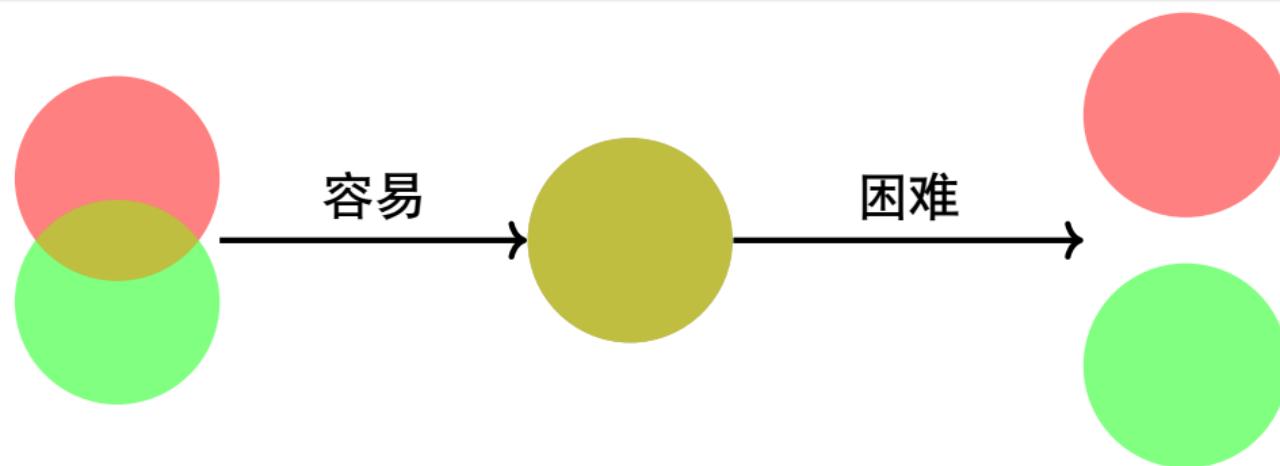
4 椭圆曲线密码

单向函数

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 定义域的任一 x , 容易计算 $y = f(x)$;
- ② 对于几乎所有属于 f 值域的任一 y , 求得 x 使 $y = f(x)$, 在计算上不可行。



离散对数问题 (Discrete Logarithm Problem, DLP)

- 给定素数 p 和整数 $a \in \{1, 2, \dots, p - 1\}$ 。
- 若给定整数 x , 求 $y = a^x \bmod p$ 很容易。
- 但是若给定 y 求 x , 则为离散对数问题。
- 目前最快方法需要

$$L(p) = \exp\{(\ln p)^{1/3}(\ln \ln p)^{2/3}\}$$

次运算。

- 例如, 当 $p = 512$ 位时, $L(p) \approx 2^{256} \approx 10^{77}$, 计算上不可行。
- The discrete logarithm problem is believed to be extremely hard, and no efficient solution is known at this point.

因数分解问题 (Factoring 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 $\exp\{c\sqrt{\ln n \ln \ln n}\}$ 次运算, 其中 c 为大于 1 的正整数。
- The problem of computing $\phi(n)$ is equivalent to factoring n , in that an efficient algorithm for one problem implies an efficient algorithm for the other.
- It remains an interesting open problem to relate (in either direction) the hardness of the discrete logarithm problem to that of the factoring problem.

指数函数

- 令 $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$, p 为素数, 则 \mathbb{Z}_p^* 及其上面的模 p 乘法运算构成有限群 $G = (\mathbb{Z}_p^*, \times_p)$ 。
- 令 $g \in \mathbb{Z}_p^*$ 为群中任意元素, $g^x \bmod p$ 称为**指数函数**。
- 称序列 $\langle g \rangle \triangleq \{g^0, g^1, g^2, \dots\}$ 为 g 产生的序列。
- 因为 \mathbb{Z}_p^* 是有限群, $\langle g \rangle$ 必重复, 为周期序列。
- 当存在最小正整数 T , 使得 $g^T \equiv 1 \pmod{p}$ 时, 称 T 为 g 在 \mathbb{Z}_p^* 中的**阶** (也称为**序**或**周期**), 记为 $\text{order}(g) = T$ 。
- 根据 Lagrange 定理, 循环群中子群的阶必定整除群的阶。
- 因此, \mathbb{Z}_p^* 中元素 g 的阶必定整除 $p-1$, 即 $\text{order}(g) \mid \text{order}(G)$

指数函数的特性：周期性 ($a^x \bmod 19$)

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

定义 (本原元)

若 $g \in \mathbb{Z}_p^*$ 的阶为 $\text{order}(g) = p - 1$, 则称 g 为模 p 的本原元。

- 本原元也称为素根或原根。
- 当 g 为模 p 运算的本原元时, 由 g 产生的序列 $\langle g \rangle$ 具有最大周期 (安全性高)。
- 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, 则 $g^a \bmod p$ 也为模 p 本原元。
- 模 p 的本原元个数为 $\phi(p - 1)$ 。

指数函数的特性：本原元

例 (本原元)

- $p = 11, g = 2, \phi(p - 1) = \phi(10) = 4$, 即存在 4 个模 11 本原元。
- 若 $g = 2$ 为模 p 本原元, 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 本原元。
- 找到一个本原元后可以很容易找到所有本原元, 问题是如何找到第一个本原元。

快速指数运算

- 如果 x 是一个大整数, 如何快速计算 g^x ? 需 $x - 1$ 次乘法。
- 将 x 写为二进制形式:

$$x = (x_{n-1} \cdots x_0) = \sum_{i=0}^{n-1} x_i 2^i = \sum_{i: x_i=1} 2^i$$

所以

$$g^x = g^{\sum_{i: x_i=1} 2^i} = \prod_{i: x_i=1} g^{2^i}$$

- $g^{11} = g^{(1011)} = g^8 \cdot g^2 \cdot g$, $g^{23} = g^{(10111)} = g^{16} \cdot g^4 \cdot g^2 \cdot g$
- 需要 $n - 1$ 次平方及 $w(x) - 1$ 次乘法, $w(x)$ 为 x 二进制串中 1 的个数。
- 平均而言, $w(x) = n/2$, 因此平均需要 $1.5n - 2$ 次乘法。

快速指数运算

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

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- 素性测试
- 单向函数与指数函数
- **计算乘法逆元**
- 中国余数定理

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

计算乘法逆元

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

例

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

目录

1 数论基础

- 数论简介
- 费马定理和欧拉定理
- 素性测试
- 单向函数与指数函数
- 计算乘法逆元
- **中国余数定理**

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

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三七七數之賸一問物幾何
答曰一十三
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

知不足齋

举例：如何由余数重构整数

例

- $\mathbb{Z}_{10} = \{0, 1, \dots, 9\}$ 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 \mathbb{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, \dots, n_k 两两互素, $n = \prod_{i=1}^k n_i$, 则以下两种表述等价:

表述一

\mathbb{Z}_n 中的任一整数 $a \in \mathbb{Z}_n$ 都对应一个 k 元组 (x_1, \dots, x_k) , 其中 $x_i = a \pmod{n_i}$, $i = 1, \dots, k$ 。

表述二

一元线性同余方程组

$$\begin{cases} x \equiv x_1 \pmod{n_1} \\ x \equiv x_2 \pmod{n_2} \\ \dots \\ x \equiv x_k \pmod{n_k} \end{cases}$$

在 \mathbb{Z}_n 中有一个公共解 x 。

CRT 的作用

- 模数 n 很大时, 模 n 的运算可以转换为模较小的数 n_i 上的运算, 事先需分解 $n = n_1 \times \cdots \times n_k$ 。
- \mathbb{Z}_n 中的算术运算可以转换为 k 元组上的算术运算。若

$$A \leftrightarrow (a_1, \dots, a_k)$$

$$B \leftrightarrow (b_1, \dots, b_k)$$

则

$$(A + B) \bmod n \leftrightarrow ((a_1 + b_1) \bmod n_1, \dots, (a_k + b_k) \bmod n_k)$$

$$(A \times B) \bmod n \leftrightarrow ((a_1 \times b_1) \bmod n_1, \dots, (a_k \times b_k) \bmod n_k)$$

因为

$$(A \cdot B) \bmod n_i = (A \bmod n_i \cdot B \bmod n_i) \bmod n_i = (a_i \cdot b_i) \bmod n_i$$

其中 $\cdot \in \{+, \times\}$ 。

CRT 的证明

当 $k = 2$ 时

已知

$$\begin{cases} x \equiv x_1 \pmod{n_1} \\ x \equiv x_2 \pmod{n_2} \end{cases}$$

其中 n_1 和 n_2 互素且 $n = n_1 n_2$, 求 $x \in \mathbb{Z}_n$ 。

- 由扩展欧几里得算法可得整数 m_1, m_2 且 $m_1 n_1 + m_2 n_2 = 1$
- 一元线性同余方程组的解为 $x = (x_1 m_2 n_2 + x_2 m_1 n_1) \pmod{n}$
- 因为

$$x \pmod{n_1} = x_1 m_2 n_2 \pmod{n_1} = x_1$$

$$x \pmod{n_2} = x_2 m_1 n_1 \pmod{n_2} = x_2$$

CRT 的证明

推论

如果

$$\begin{cases} x \equiv y \pmod{p} \\ x \equiv y \pmod{q} \end{cases}$$

其中 p 和 q 互素，那么

$$x \equiv y \pmod{pq}$$

CRT 的证明

当 $k = 3$ 时

已知

$$\begin{cases} x \equiv x_1 \pmod{n_1} \\ x \equiv x_2 \pmod{n_2} \\ x \equiv x_3 \pmod{n_3} \end{cases}$$

其中 n_1, n_2, n_3 两两互素且 $n = n_1 n_2 n_3$, 求 $x \in \mathbb{Z}_n$ 。

- $k > 2$ 时的情况可以归约为 $k = 2$ 时的情况。
- 由前两个等式可以确定 $x \equiv x_{12} \pmod{n_1 n_2}$ 。
- 再与第三个等式可以确定 $x \equiv x_{123} \pmod{n_1 n_2 n_3}$ 。

CRT 的证明

完整证明.

- $\forall i, n/n_i$ 与 n_i 互素 $\Rightarrow \exists y_i, (n/n_i)y_i \bmod n_i = 1$ 。
- $\forall i \neq j, n/n_i$ 有因子 $n_j \Rightarrow (n/n_i)y_i \bmod n_j = 0$ 。
- 令

$$x \triangleq \sum_{i=1}^k \frac{n}{n_i} y_i x_i \bmod n$$

因为

$$x \bmod n_j = \frac{n}{n_j} y_j x_j \bmod n_j = x_j$$

所以 x 是 $x \bmod n_j = x_j, j = 1, \dots, k$ 的公共解。



“物不知其数”问题求解

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

$$n_1 = 3, n_2 = 5, n_3 = 7$$

$$x_1 = 2, x_2 = 3, x_3 = 2$$

$$n = 3 \times 5 \times 7 = 105$$

- 求 $y_i = (n/n_i)^{-1} \bmod n_i$, 得

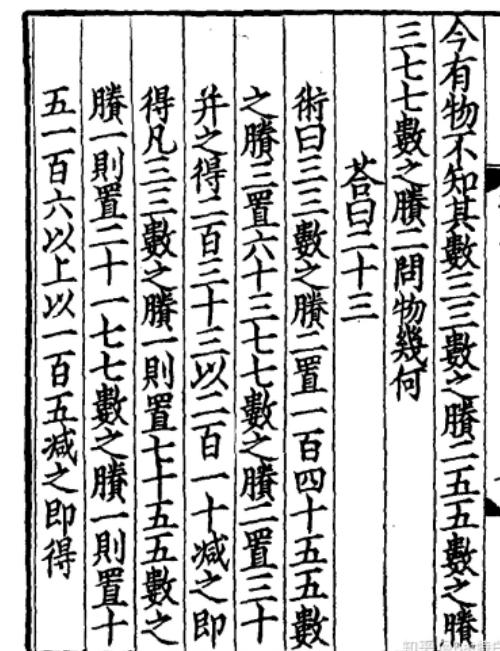
$$y_1 = 2, y_2 = 1, y_3 = 1$$

- 代入前面得到的公式中

$$x = \sum_i \frac{n}{n_i} y_i x_i \bmod n$$

$$\begin{aligned}
 &= (70 \times 2 + 21 \times 3 + 15 \times 2) \bmod 105 \\
 &= 23
 \end{aligned}$$

从而得到结果 23。



“物不知其数”问题求解

《孙子歌诀》

三人同行七十稀，
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》

目录

1 数论基础

2 基本概念与 RSA 算法

- 公钥密码学的基本原理
- RSA 非对称加密算法
- RSA 的安全性分析

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

目录

1 数论基础

2 基本概念与 RSA 算法

- 公钥密码学的基本原理
- RSA 非对称加密算法
- RSA 的安全性分析

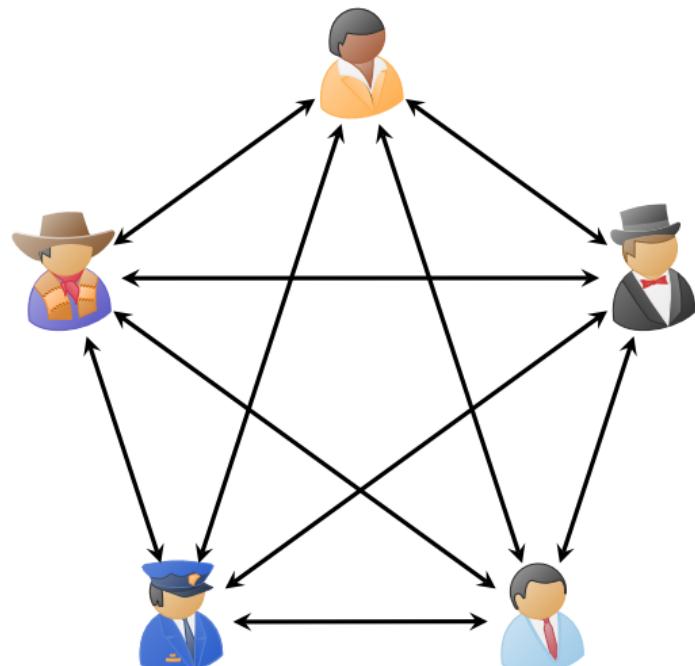
3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

对称密码体制的问题

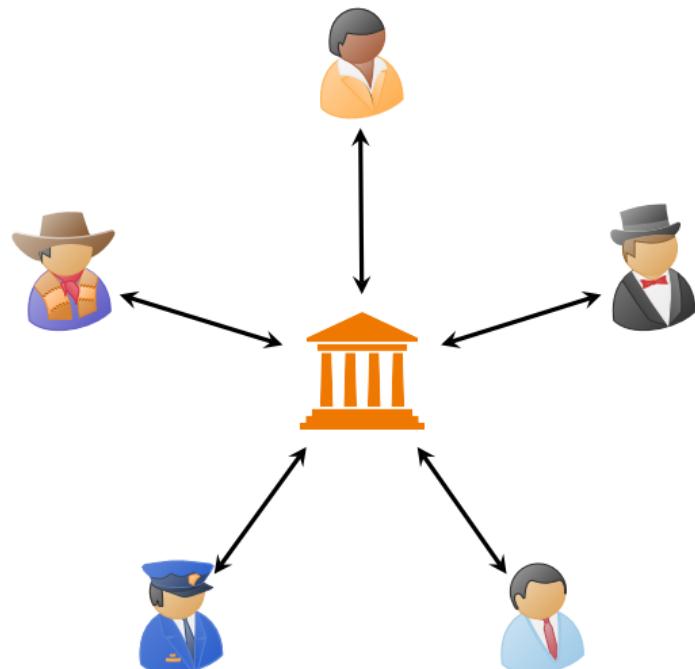
- 加密能力与解密能力捆绑在一起。
- **密钥分发困难**：密钥更换、传递和交换需要可靠信道。
- **密钥管理困难**：无法满足陌生人之间通信的保密要求。
- **身份认证问题**：难以保障可信通信。

对称密码体制的问题



- n 个人相互通信需要管理 $\binom{n}{2}$ 对密钥
- 如何在任意两个人之间分发只有他们自己知道的密钥？

对称密码体制的问题



- 使用第三方中央可信服务器可以减少密钥数量
- 通信/计算瓶颈问题？
- 是否真实存在可信第三方中央服务器？

Whitefield Diffie 和 Martin Hellman

644

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation

Stanford | News

[Home](#)[Find Stories](#)[For Journalists](#)[Contact](#)

Stanford Report, March 1, 2016

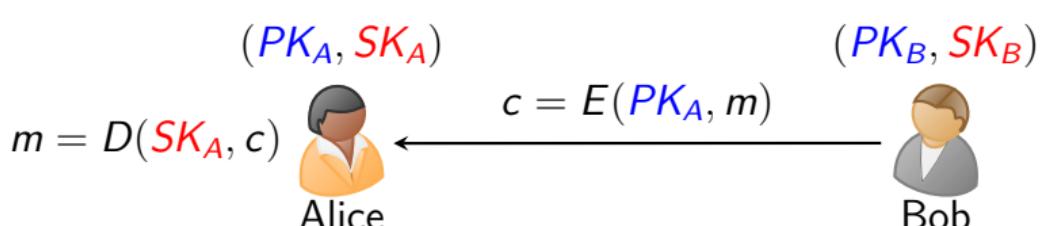
Stanford cryptography pioneers Whitfield Diffie and Martin Hellman win ACM 2015 A.M. Turing Award

The groundbreaking algorithm from Whitfield Diffie and Martin Hellman enabled a secure Internet and sparked a clash with the NSA that foreshadowed current privacy battles between government agencies and Silicon Valley companies.



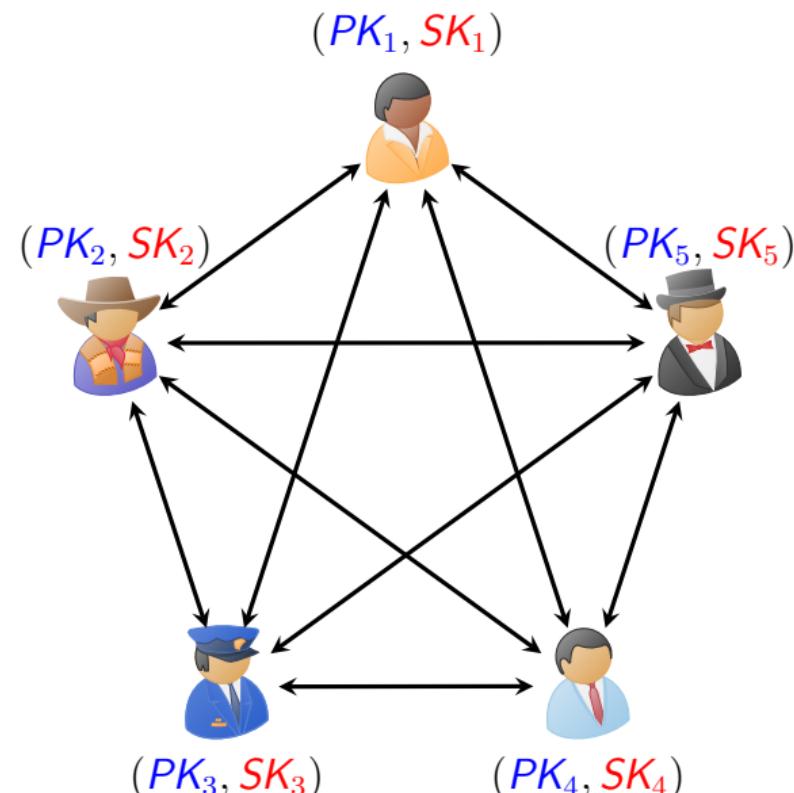
W. Diffie 和 M. Hellman 提出的新设想

- 每个用户 A 有一个加密密钥 PK_A ，一个解密密钥 SK_A 。
- 解密密钥 SK_A 需要保密，而加密密钥 PK_A 可以公开，要求 PK_A 的公开不影响 SK_A 的安全。
- 若用户 B 要向用户 A 秘密发送明文 m ，可查询 A 的公开密钥 PK_A ，加密后得到密文 $c = E(PK_A, m)$ 。
- 用户 A 收到密文 c 后，用只有用户 A 才拥有的解密密钥 SK_A 对 c 进行解密得到明文 $m = D(SK_A, c)$ 。



非对称密码体制的基本特点

- 加密能力与解密能力分开。
- 密钥分发简单, n 个用户只需要 $2n$ 个密钥。
- 可以满足陌生人之间的保密通信。
- 可以实现数字签名。

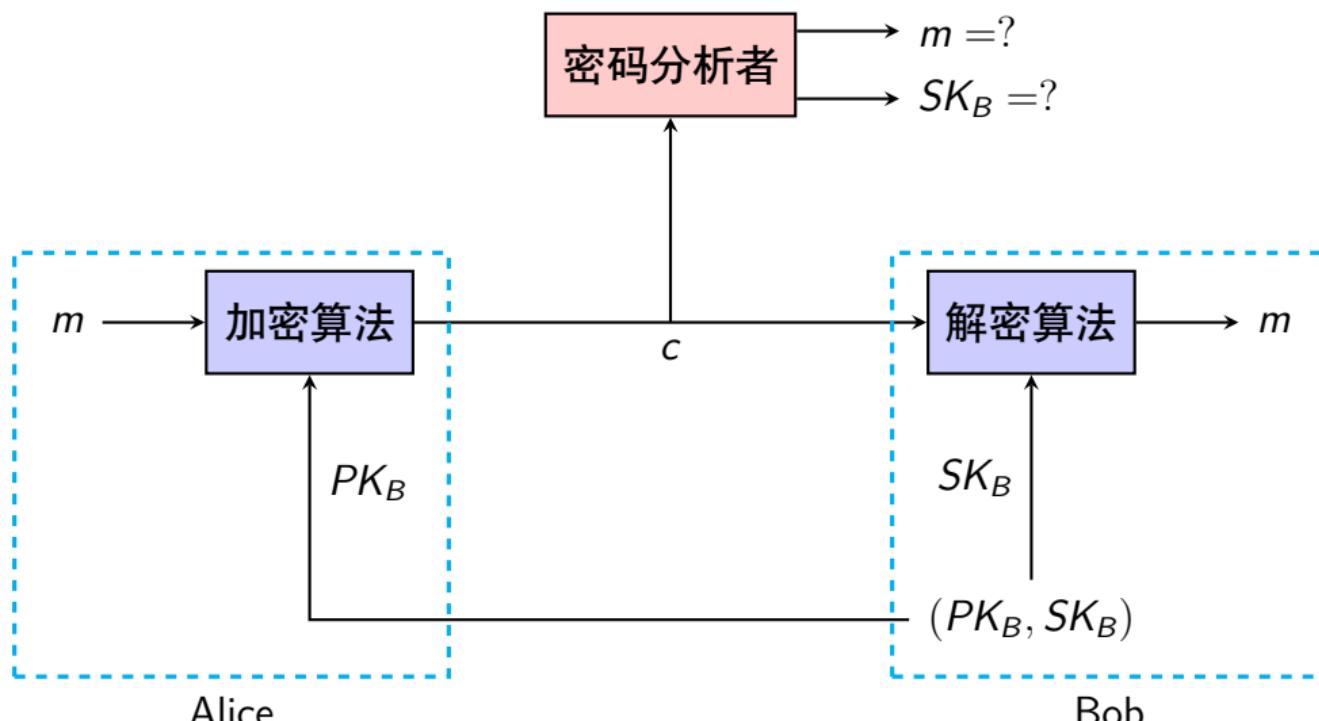


公钥密码体制

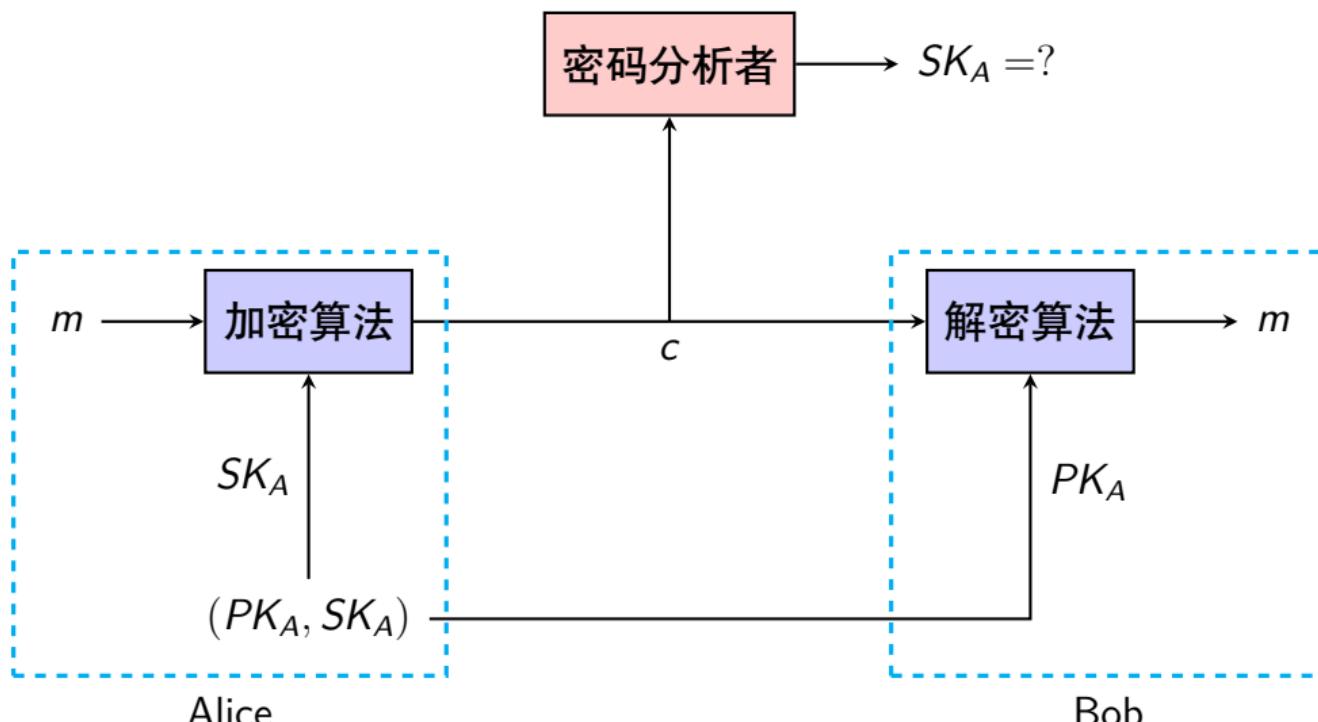
公钥密码体制的组成

- **明文**：算法的输入，可读信息或数据。
- **加密算法**：对明文进行转换。
- **公钥和私钥**：算法的输入，分别用于加密和解密。
- **密文**：算法的输出，依赖于明文和密钥。
- **解密算法**：根据密文和密钥，还原明文。
- 公钥算法依赖于一个**加密密钥**和一个与之相关的不同的**解密密钥**。算法有如下特点：
 - 仅由密码算法和加密密钥来确定解密密钥在计算上不可行
 - 两个密钥的任何一个都可用来加密，另一个用来解密

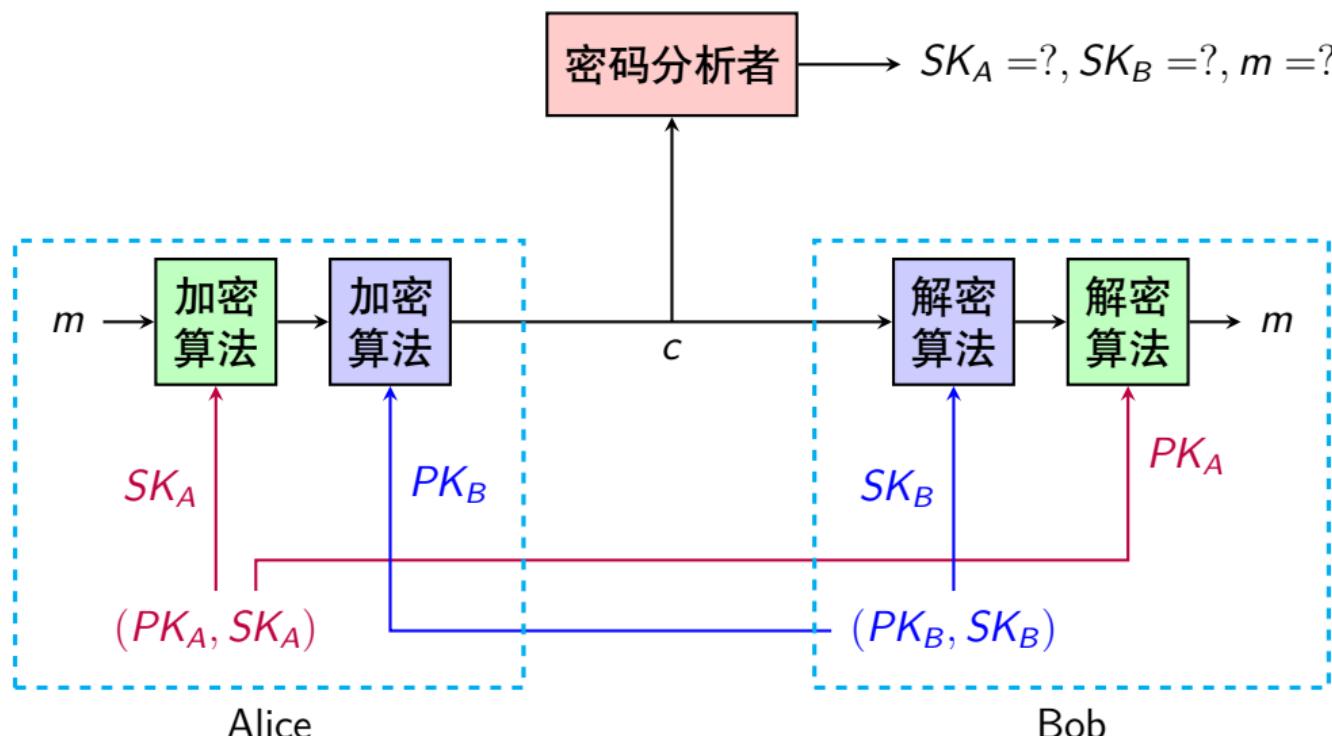
公钥密码：实现保密



公钥密码：实现认证



公钥密码：同时实现保密与认证



公钥密码体制的应用

- **加密/解密**: 发送方用接收方的公钥对消息加密
- **数字签名**: 发送方用其私钥对消息签名, 可以对整体消息签名或对消息的摘要签名
- **密钥交换**: 通信双方交换会话密钥

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

公钥密码体制的要求

- 容易产生一对密钥（公钥 PK 和私钥 SK ）。
- 不难计算 $c = E(PK, m)$ 和 $m = D(SK, c)$ 。
- 知道 PK , 计算 SK 不可行。
- 不知道 SK , 即使知道 PK, E, D 及 c , 计算 m 不可行。
- 对明文 $m, E(PK, m)$ 有定义, 且 $D(SK, E(PK, m)) = m$ 。
- 对密文 $c, D(SK, c)$ 有定义, 且 $E(PK, D(SK, c)) = c$ 。
- 两个密钥可以交换顺序, 即

$$D(PK, E(SK, m)) = D(SK, E(PK, m))$$

公钥密码体制的分析

- **穷举攻击**：公钥密码易受穷举攻击，解决方法是使用长密钥。同时为了便于实现加密和解密，又希望密钥足够短。目前公钥密码仅限于密钥管理和签名。
- **从给定的公钥计算出私钥**：尚未在数学上证明对一特定公钥算法这种攻击是不可行的。因此包括 RSA 在内的任何算法都是值得怀疑的。
- **穷举消息攻击**：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。

目录

1 数论基础

2 基本概念与 RSA 算法

- 公钥密码学的基本原理
- RSA 非对称加密算法
- RSA 的安全性分析

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

RSA 非对称加密算法

- 1977 年，Ron Rivest、Adi Shamir、Len Adleman 提出了非对称加密算法 RSA，基于**大合数的素因子分解**难题。
- 1994 年 4 月一个小组通过 Internet 合作，8 个月时间成功分解 129 位的数，大约 428 比特；1999 年分解 155 位合数，最新的记录是 2005 年 5 月分解 200 位十进制数。
- RSA 专利于 2000 年 9 月 20 日到期。



Adi Shamir

- 1952 年出生于以色列，现任以色列魏兹曼科学研究院计算机科学与应用数学系教授。
- Adi Shamir 是信息加密和解密领域的顶尖专家。他是 RSA 加密算法的开发者之一，该方法改变了世界计算机通信的面貌，是电子商务和信息安全的基本支柱。



We are proud to announce
The 2024 Wolf Prize
Laureate in Mathematics

Adi Shamir

Israel

Prof. Shamir has been recognized
"for his fundamental contributions
to Mathematical Cryptography"

RSA 密码体制算法流程

Alice 按照以下步骤生成自己的公私钥

- ① 随机选择两个秘密大素数 p 和 q , $p \neq q$;
- ② 计算公开模数 $n = pq$ 及秘密欧拉函数 $\phi(n) = (p - 1)(q - 1)$;
- ③ 选择一个小于 $\phi(n)$ 且与 $\phi(n)$ 互素的数作为公钥 e ;
- ④ 计算私钥 $d = e^{-1} \bmod \phi(n)$;
- ⑤ 公开 n, e 。

3、4 步中的 e 和 d 可以交换

加密及解密

- Bob 用公开的 n 和 e 对消息 $m \in \mathbb{Z}_n$ 加密: $c = m^e \bmod n$
- Alice 利用自己的私钥 d 对密文 $c \in \mathbb{Z}_n$ 解密: $m = c^d \bmod n$

RSA 解密正确性的推导

- 如果按照规定的方式加密，则

$$m = c^d \pmod{n} = m^{ed} \pmod{n}$$

要能正确解密须说明 $m^{ed} \equiv m \pmod{n}$ 。

- 由于 $n = pq$ ，利用 CRT 的推论：

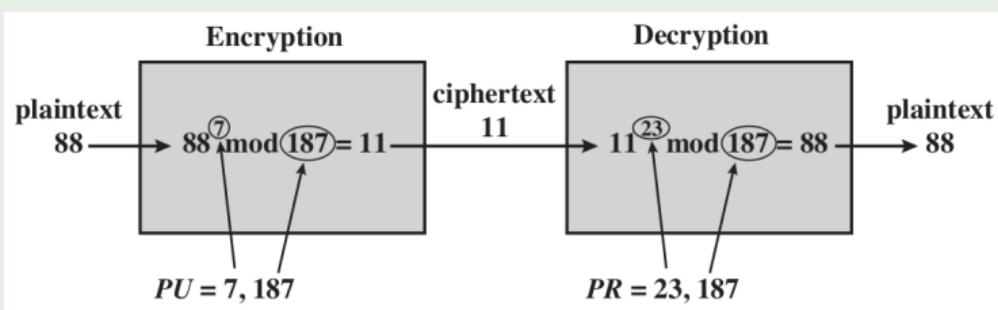
如果 $\begin{cases} m^{ed} \equiv m \pmod{p} \\ m^{ed} \equiv m \pmod{q} \end{cases}$ ，则有 $m^{ed} \equiv m \pmod{n}$

- 由于 $ed \equiv 1 \pmod{\phi(n)}$ ，则

$$\begin{aligned} m^{ed} &\equiv m^{k\phi(n)+1} \equiv m^{k'(p-1)+1} \equiv m^{k'p-k'+1} \\ &\equiv m^{k'p}m^{-k'+1} \equiv m^{k'}m^{-k'+1} \equiv m \pmod{p} \end{aligned}$$

同理可证 $m^{ed} \equiv m \pmod{q}$ 。

RSA 算法举例



- 选择 $p = 17, q = 11$, 则 $n = pq = 187, \phi(n) = 160$;
- 选择 $e = 7$ 满足 $\gcd(7, 160) = 1, d = 23$;
- 公钥 $PK = 7$, 私钥 $SK = 23$;
- 明文 $m = 88$;
- 加密计算 $c = 88^7 \text{ mod } 187 = 11$;
- 解密计算 $m = 11^{23} \text{ mod } 187 = 88$ 。

目录

1 数论基础

2 基本概念与 RSA 算法

- 公钥密码学的基本原理
- RSA 非对称加密算法
- RSA 的安全性分析

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

能否由公钥计算私钥？

- 第三方密码分析者如果想要解密密文 c ，需要知道 Alice 的私钥 d 。
- RSA 算法中模数 n 和公钥 e 公开，第三方密码分析者能否由公开信息计算出 Alice 的私钥 d ？
- 计算私钥 d 需要计算 $d = e^{-1} \bmod \phi(n)$ ，但是 $\phi(n)$ 未公开。
- 计算 $\phi(n)$ 的难度等同于对模数 n 进行因数分解，但是因数分解问题 (FAC) 属于单向函数，当 n 足够大时，目前没有高效求解算法。
- 因此 RSA 算法的安全性依赖于单向函数 FAC 难题的求解。

数学攻击

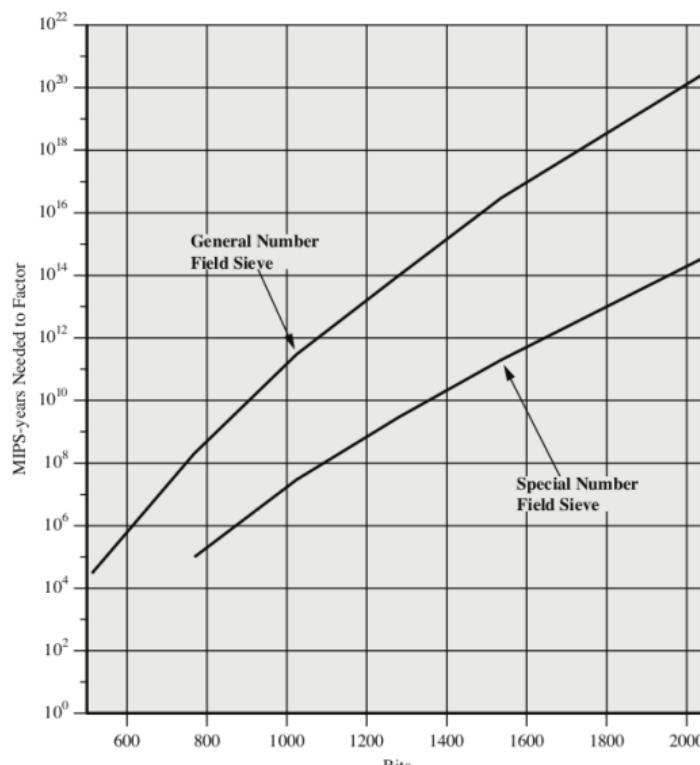
- 1977 年, RSA 的三位发明者在《科学美国人》杂志上发布一段密文让读者解密, 解得明文者可获得 100 美元, 他们预言需要 4×10^{16} 年才能解得明文。
- 这里 n 为 129 位十进制位, 或 428 位二进制位。
- 但是, 一个在互联网上工作的小团体只用了 8 个月的时间, 于 1994 年 4 月正确解密。
- RSA 实验室也发布了使用不同 n 长度加密的密文, 让公众解密。

数学攻击

Table 9.5 Progress in RSA Factorization

Number of Decimal Digits	Number of Bits	Date Achieved
100	332	April 1991
110	365	April 1992
120	398	June 1993
129	428	April 1994
130	431	April 1996
140	465	February 1999
155	512	August 1999
160	530	April 2003
174	576	December 2003
200	663	May 2005
193	640	November 2005
232	768	December 2009

MIPS-years Needed to Factor



- MIPS: million instructions per second
- MIPS-year: the number of instructions executed during one year of computing at one MIPS.
- GNFS 和 SNFS 是两种大数分解算法

数学攻击

- n 的位数应取 1024 到 2048 位；
- p 和 q 的长度应仅相差几位， p 和 q 都应约在 10^{75} 到 10^{100} 之间；
- $(p - 1)$ 和 $(q - 1)$ 都应有一个大的素因子；
- $\gcd(p - 1, q - 1)$ 应该较小。

目录

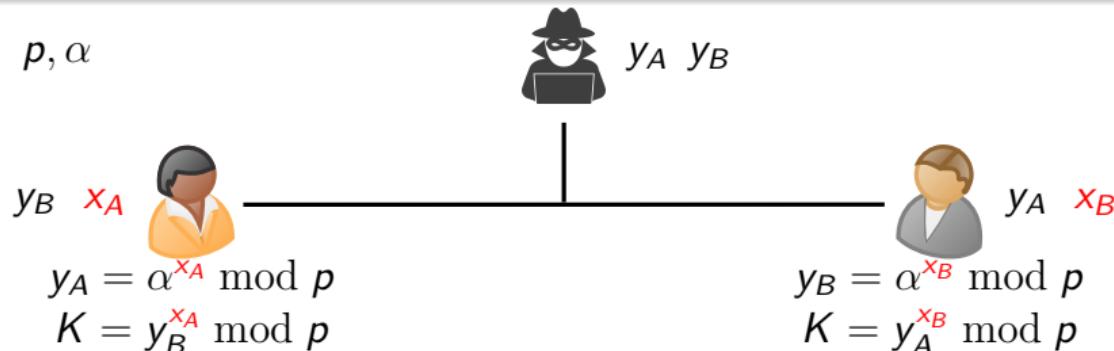
- 1 数论基础
- 2 基本概念与 RSA 算法
- 3 Diffie-Hellman 密钥交换协议
- 4 椭圆曲线密码

Diffie-Hellman 密钥交换协议

- DH 密钥交换算法是一种密钥分发机制，不是用于加密消息。
- 由通信双方公私钥生成双方共享的会话密钥。
- DH 密钥交换算法的安全性依赖于求解离散对数问题：
 - 已知 g, p, y ，计算满足 $y = g^x \bmod p$ 的 x 。
 - 计算时间复杂度为 $\exp\{(\ln p)^{1/3}(\ln \ln p)^{2/3}\}$ 。

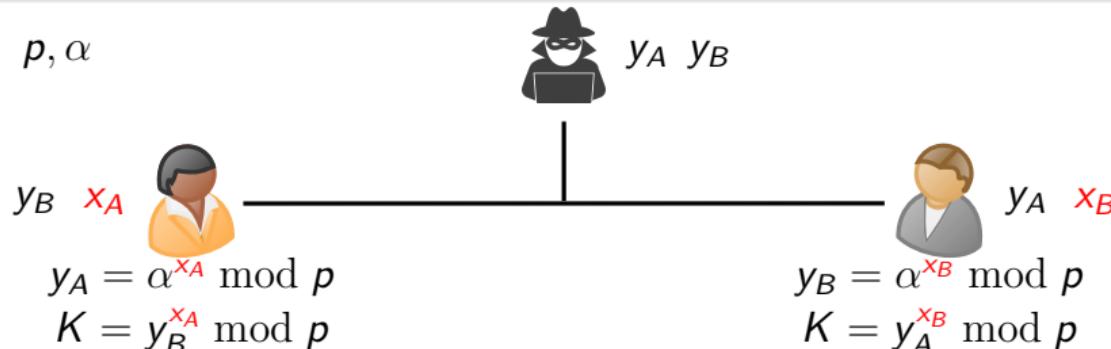
p 的比特位数	计算量	按 $1\mu s$ 计算一次需要的计算时间
200	2.7×10^{11}	$2 \sim 3$ 天
664	1.2×10^{23}	10^{12} 天 ≈ 2.7 亿年

Diffie-Hellman 密钥交换协议



- Alice 和 Bob 选择大素数 p 和它的一个素根 α ;
- Alice 选择一个随机数 x_A 作为自己的私钥, 并计算公钥 $y_A = \alpha^{x_A} \pmod{p}$, 并将 y_A 发送给 Bob;
- Bob 选择一个随机数 x_B 作为自己的私钥, 并计算公钥 $y_B = \alpha^{x_B} \pmod{p}$, 并将 y_B 发送给 Alice;
- Alice 计算 $K = y_B^{x_A} \pmod{p}$ 作为会话密钥;
- Bob 计算 $K = y_A^{x_B} \pmod{p}$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



Alice 计算 K

$$K \equiv y_B^{x_A} \equiv \alpha^{x_B x_A} \pmod{p}$$

Bob 计算 K

$$K \equiv y_A^{x_B} \equiv \alpha^{x_A x_B} \pmod{p}$$

- Eve 虽然知道 y_A, y_B , 但无法获得 x_A 或 x_B , 所以无法得到 K 。
- Eve 由 $y_A = \alpha^{x_A} \pmod{p}$ 计算 x_A 是一个离散对数问题。
- Eve 由 $y_B = \alpha^{x_B} \pmod{p}$ 计算 x_B 也是一个离散对数问题。

Diffie-Hellman 密钥交换协议

Alice

Alice and Bob share a prime number q and an integer α , such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$

Bob

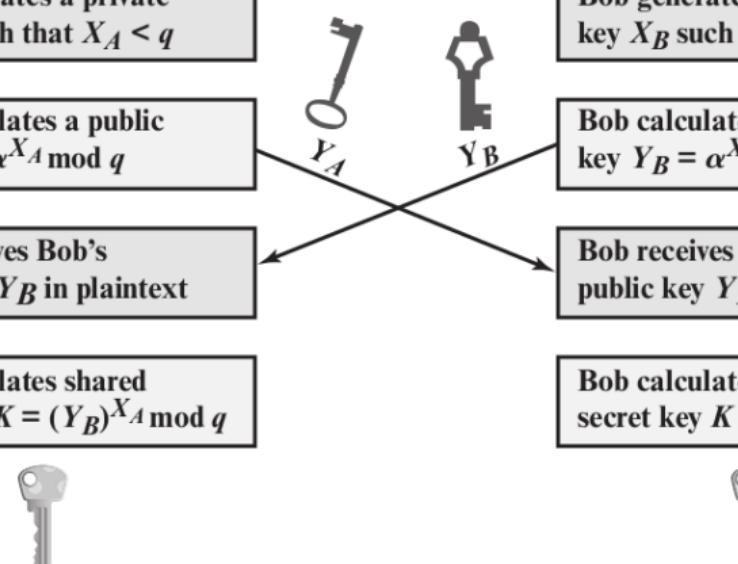
Alice and Bob share a prime number q and an integer α , such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

Bob receives Alice's public key Y_A in plaintext

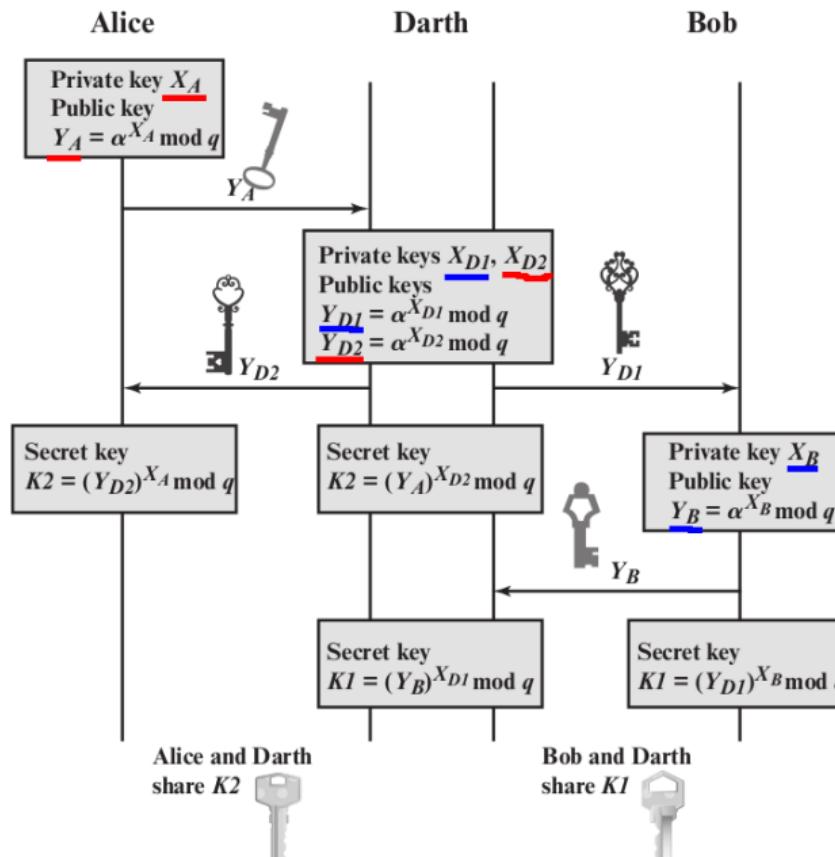
Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$



Diffie-Hellman 密钥交换举例

- Users Alice & Bob who wish to swap keys;
- Agree on prime $p = 353$ and $\alpha = 3$;
- Select random secret keys:
 - A chooses $x_A = 97$,
 - B chooses $x_B = 233$.
- Compute public keys:
 - $y_A = 3^{97} \bmod 353 = 40$ (Alice)
 - $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Compute shared session key as:
 - $K = y_B^{x_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
 - $K = y_A^{x_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

中间人攻击 Man-in-the-middle Attack



注意

密钥交换协议不能抵抗上述攻击，因为它未对通信参与方进行认证，可通过数字签名克服。

目录

1 数论基础

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

- 椭圆曲线算术（实数域）
- 有限域上的椭圆曲线
- 椭圆曲线密码学

目录

1 数论基础

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

- 椭圆曲线算术 (实数域)
- 有限域上的椭圆曲线
- 椭圆曲线密码学

椭圆曲线算术 (实数域)

- 椭圆曲线是由威尔斯特拉斯方程所确定的平面曲线：

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

考虑其标准形式：

$$y^2 = x^3 + ax + b$$

称满足上述方程的序偶 (x, y) 为椭圆曲线 $E(a, b)$ 上的点。

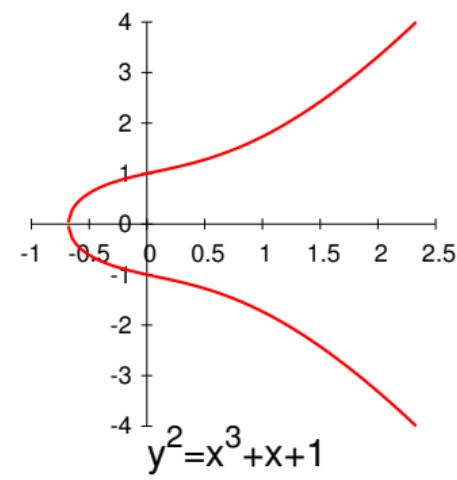
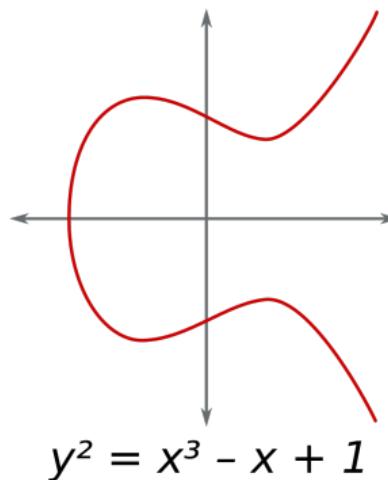
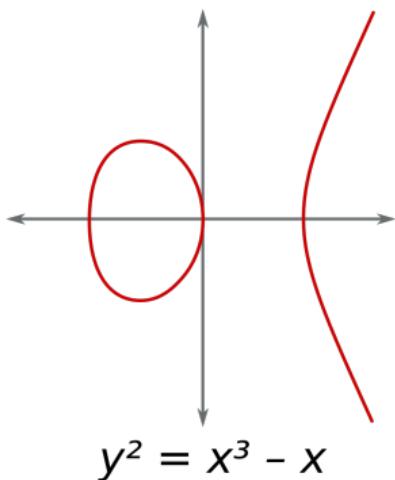
- 要求 a, b 满足条件 $4a^3 + 27b^2 \neq 0$ 使 $x^3 + ax + b = 0$ 不含重根；否则椭圆曲线在重根处不存在切线。

$E(0,0)$ 在
(0,0) 点不
光滑

$E(-3,2)$ 在
(1,0) 点有
交叉

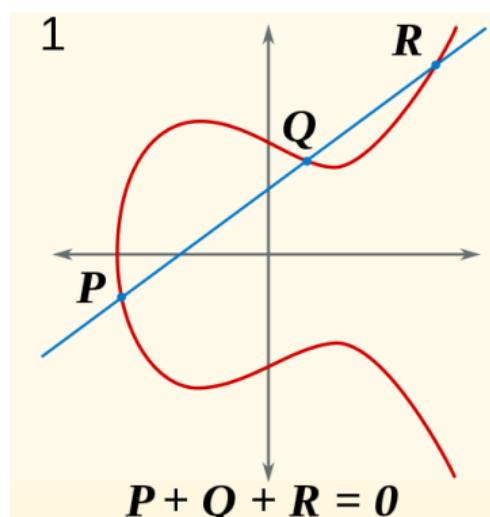
97 / 122

椭圆曲线举例



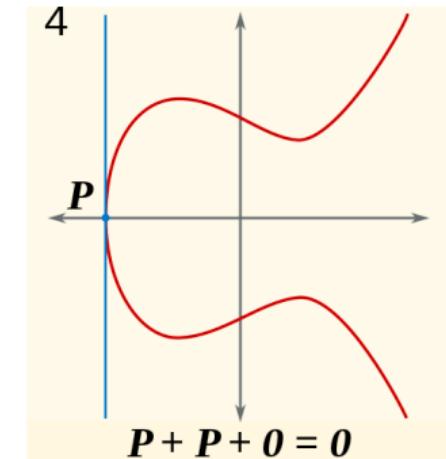
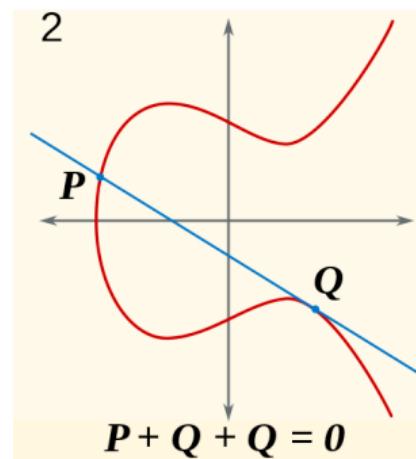
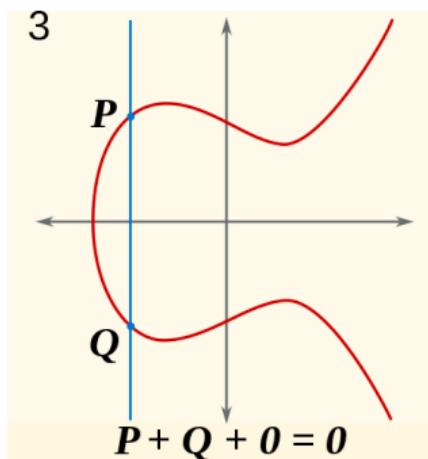
椭圆曲线上形式加法的定义

- 椭圆曲线包括一个称为无穷远点或零点的元素, 记为 O ;
- O 是加法的零元, 对于椭圆曲线上的任意点 P , 满足 $P + O = O + P = P$;
- 如果椭圆曲线上三个点处于一条直线上, 那么它们的和为 O ;



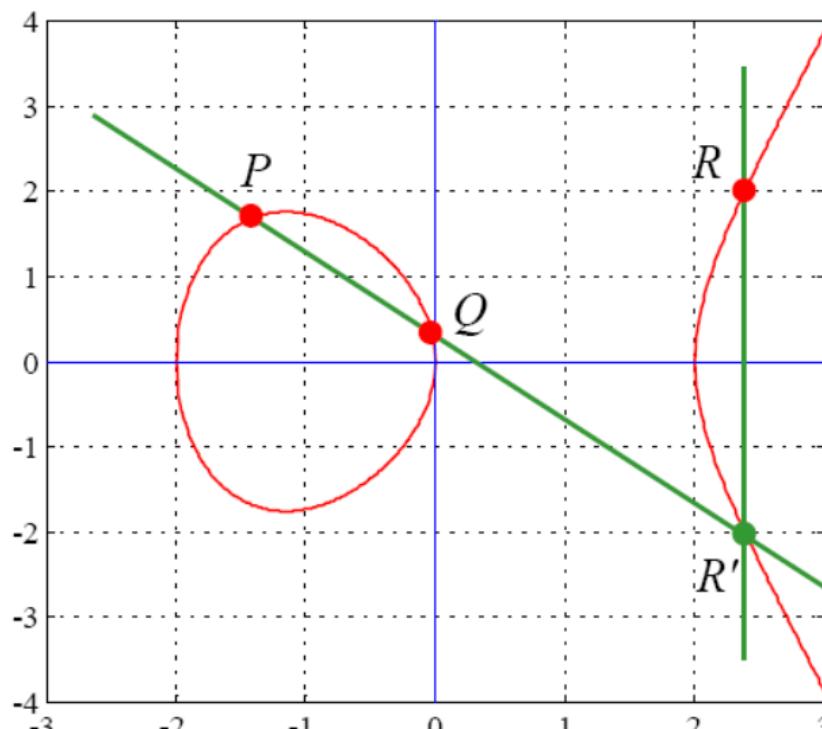
椭圆曲线上形式加法的定义

- 一条垂直线与曲线相交于 $P = (x, y)$ 和 $Q = (x, -y)$ ，也相交于无穷点 O ，有 $P + Q + O = O$ ，称 $Q = -P$ 为 P 的**负元**；
- 在点 Q 处画一切线求出另一交点 P ，则 $Q + Q + P = O$ ，即 $2Q = -P$ ；
- 椭圆曲线上的点及其上的形式加法构成一个 Abel 群。



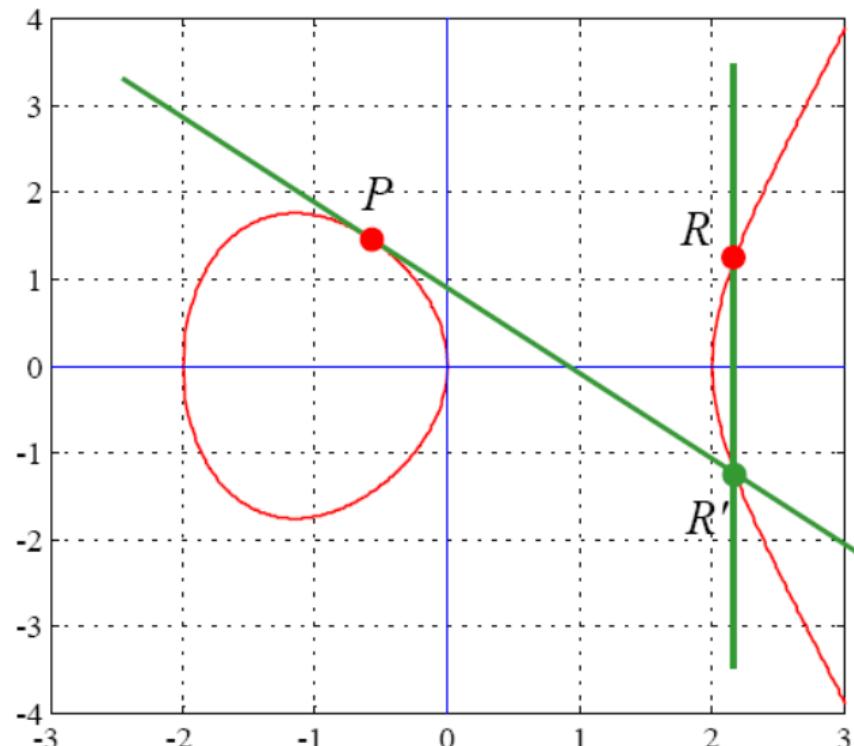
加法

$$R = P + Q \quad (\text{或 } R = P \cdot Q)$$



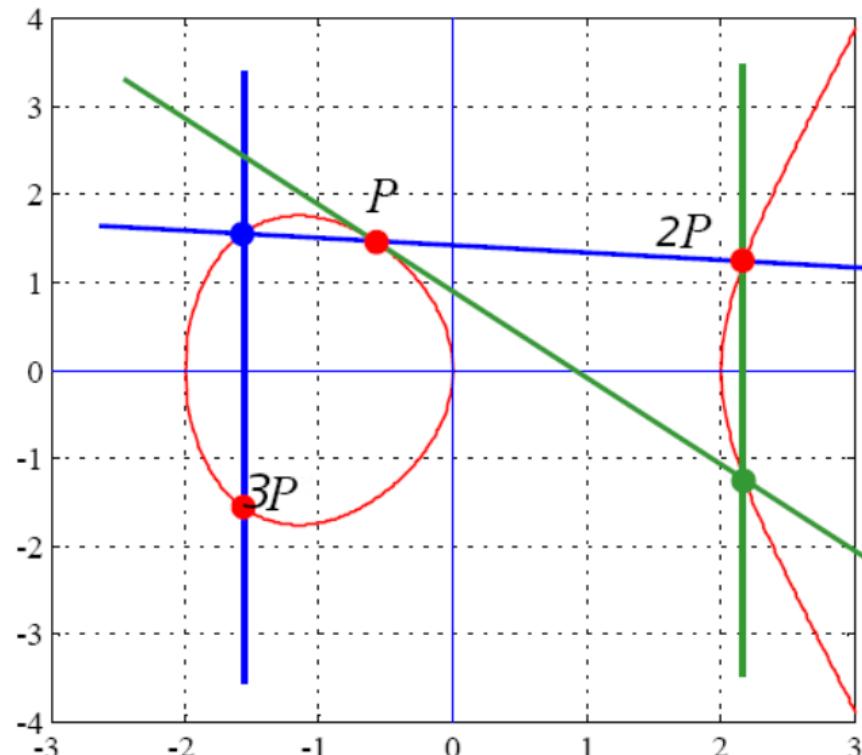
累加

$$R = P + P = 2P \quad (\text{或 } R = P^2)$$



累加

$$R = P + P + P = 3P \quad (\text{或 } R = P^3)$$



计算直线与椭圆曲线交点

- 经过点 P 和 Q 的直线:

$$y = sx + y_0, \text{ 其中}$$

$$s = \frac{y_Q - y_P}{x_Q - x_P}$$

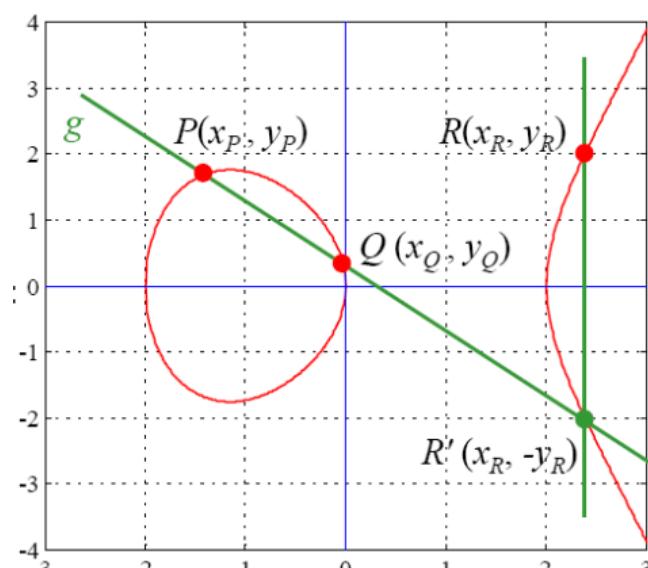
$$y_0 = y_P - sx_P$$

- 直线与曲线的另一个交点:

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 R 点坐标:

$$\begin{cases} x_R = s^2 - x_P - x_Q \\ y_R = -(sx_R + y_0) \end{cases}$$



计算切线与椭圆曲线交点

- P 点的切线: $y = sx + y_0$, 其中

$$s = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

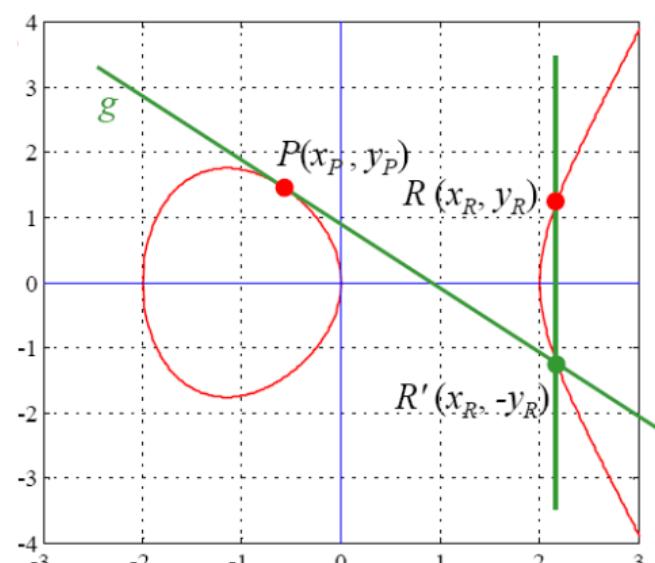
$$y_0 = y_P - sx_P$$

- 切线与曲线交点:

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 R 点坐标:

$$\begin{cases} x_R = s^2 - 2x_P \\ y_R = -(sx_R + y_0) \end{cases}$$



目录

1 数论基础

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

- 椭圆曲线算术 (实数域)
- **有限域上的椭圆曲线**
- 椭圆曲线密码学

有限域上的椭圆曲线

- 椭圆曲线密码体制使用的是变元和系数均为有限域中元素的椭圆曲线。
- 定义在 $\text{GF}(p)$ 上的椭圆曲线 $E_p(a, b)$:

$$y^2 = (x^3 + ax + b) \bmod p$$

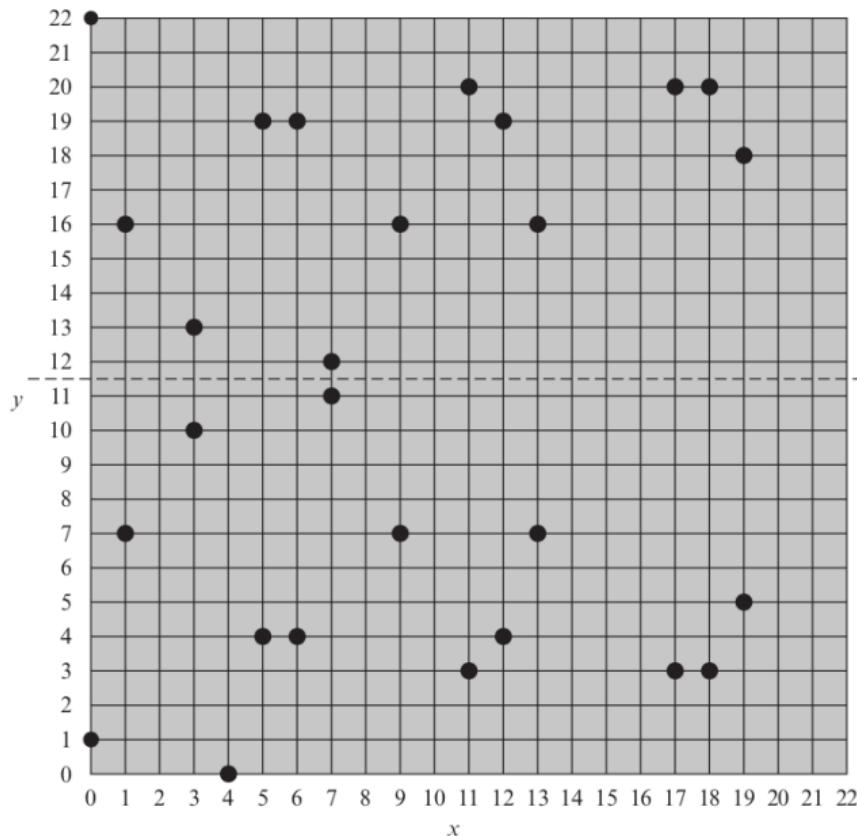
其中变元和系数均取自集合 \mathbb{Z}_p , 模 p 运算。

椭圆曲线 $E_{23}(1, 1)$ 上的点

- 对于每个 $x \in \mathbb{Z}_p$, 计算 $y^2 = x^3 + x + 1 \pmod{p}$;
- 对每个结果确定它是否有一个模 p 的平方根;
- 如果没有, 在 $E_{23}(1, 1)$ 中就没有具有这个 x 值的点;
- 如果有, 就有两个满足平方根是 y 的值 (除非这个值是单个的 y 值 0)。这些点就是 $E_{23}(1, 1)$ 上的点 (外加无穷远点)。

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

椭圆曲线 $E_{23}(1, 1)$ 上的点



椭圆曲线 $E_{11}(1, 6)$ 上的点

- 在 $GF(11)$ 上找出椭圆曲线 $y^2 = x^3 + x + 6 \bmod 11$ 的点；
- 有 12 个点，加上无穷远点 O 共有 $n = 13$ 个元素；
- n 称为椭圆曲线群的阶或序 (Order)，与参数 a, b 有关。

x	y^2	$Y_{1,2}$	$P(x, y)$	$P'(x, y)$
0	6	-		
1	8	-		
2	5	4, 7	(2, 4)	(2, 7)
3	3	5, 6	(3, 5)	(3, 6)
4	8	-		
5	4	2, 9	(5, 2)	(5, 9)
6	8	-		
7	4	2, 9	(7, 2)	(7, 9)
8	9	3, 8	(8, 3)	(8, 8)
9	7	-		
10	4	2, 9	(10, 2)	(10, 9)

椭圆曲线点加运算

- 将椭圆曲线 $E_{11}(1, 6)$ 上的点 $P = (2, 4)$ 反复累加
- 计算 $2P = P + P$

$$\begin{cases} s &= \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P} \\ y_0 &= y_P - sx_P \end{cases}$$

$$\begin{cases} x_R &= s^2 - 2x_P \\ y_R &= -(sx_R + y_0) \end{cases}$$

- 计算 $3P = P + P + P = 2P + P$

$$\begin{cases} s &= \frac{y_Q - y_P}{x_Q - x_P} \\ y_0 &= y_P - sx_P \end{cases}$$

$$\begin{cases} x_R &= s^2 - x_P - x_Q \\ y_R &= -(sx_R + y_0) \end{cases}$$

- 所有运算均在 $GF(11)$ 上进行。

椭圆曲线点加运算

- 取 $P = (2, 4)$, 计算 $2P = P + P$

$$s = \frac{3 \times 4 + 1}{2 \times 4} = \frac{13}{8} = 7 \times 2 = 3 \quad x_R = 9 - 2 \times 2 = 5$$

$$y_0 = 4 - 3 \times 2 = -2 = 9 \quad y_R = -(3 \times 5 + 9) = 9$$

所以 $2P = (5, 9)$

- 再计算 $3P = P + P + P = 2P + P$

$$s = \frac{9 - 4}{5 - 2} = \frac{5}{3} = 4 \times 5 = 9 \quad x_R = 81 - 2 - 5 = 8$$

$$y_0 = 4 - 9 \times 2 = -3 = 8 \quad y_R = -(9 \times 8 + 8) = -3 = 8$$

所以 $3P = (8, 8)$

目录

1 数论基础

2 基本概念与 RSA 算法

3 Diffie-Hellman 密钥交换协议

4 椭圆曲线密码

- 椭圆曲线算术（实数域）
- 有限域上的椭圆曲线
- **椭圆曲线密码学**

椭圆曲线密码学

- 大多数公开密钥密码系统都使用非常大的整数或多项式，计算量大，密钥和报文存储量也极大。
- 椭圆曲线密码系统可以**使用较短的密钥**实现同样的安全强度。
- 椭圆曲线的加法类似于模乘，累加类似于模指数。
- 需要有对应于 DLP 的难解问题。

椭圆曲线对数问题

- $Q = kP$, 其中 Q, P 为 $E_p(a, b)$ 上的点, 整数 $k < p$;
- 给定 k, P , 容易计算 $Q = kP$;
- 但是给定 Q, P , 求 k 很困难。

椭圆曲线对数问题

例

- 考虑椭圆曲线 $E_{23}(9, 17)$, 即 $y^2 = x^3 + 9x + 7 \pmod{23}$ 。
 $P = (16, 5)$ 和 $Q = (4, 5)$ 是椭圆曲线上的点, 且 $Q = kP$, 求 k 为多少?
- 可以通过穷举攻击方法, 多次计算 P 的倍数直至找到 Q :
 $P = (16, 5), 2P = (20, 20), 3P = (14, 14), 4P = (19, 20), 5P = (13, 10), 6P = (7, 3), 7P = (8, 7), 8P = (12, 17), 9P = (4, 5)$
- 所以 $k = 9$ 。
- 实际应用中, k 的值非常大, 穷举攻击不可行。

椭圆曲线密码

椭圆曲线密码系统

- **域标识**: 定义椭圆曲线采用的有限域椭圆曲线参数, 即系数 a 和 b 。
- **基准点** (Base Point): 指定的椭圆曲线上的点 G 。
- **阶** (Order): G 点的阶 n , 使得 $nG = O$, 记作 $order(G) = n$ 。

椭圆曲线公钥系统

- 定义在有限域上的椭圆曲线, 例如 $E_p(a, b)$
- 选择基准点 $G = (x, y)$
- 选择整数 $k < order(G)$ 作为私钥
- 公钥为 $P = kG$

椭圆曲线加密: ECC ElGamal

ECC ElGamal 加密算法

- **加密**: 发送方随机选择一个正整数 r , 加密点 P_m 产生密文 $c = \{c_1, c_2\} = \{rG, P_m + rP\}$
- **解密**: 接收方解密 $c_2 - kc_1 = P_m + rP - krG = P_m$

例 $(E_{751}(-1, 188), G = (0, 376))$

- 发送消息 $P_m = (562, 201)$, 接收方的公钥 $P = (201, 5)$ 。
- 发送方首先随机选择 $r = 386$, 并计算 $rG = (676, 558)$
- $P_m + rP = (562, 201) + 386(201, 5) = (385, 328)$
- 这样, 密文即为 $C = \{rG, P_m + rP\} = \{(676, 558), (385, 328)\}$
- 接收方用私钥 k 解密 $c_2 - kc_1 = P_m$

ECC Diffie-Hellman 密钥交换

类似于 Diffie-Hellman 密钥交换, ECC 也可以实现密钥交换:

- Alice 和 Bob 选择合适的 ECC, 例如 $E_p(a, b)$;
- 选择基准点 G , 要求 $n = \text{order}(G)$ 是一个大整数;
- Alice 和 Bob 之间的密钥交换如下:

基于 ECC 的 Diffie-Hellman 密钥交换协议

- Alice 和 Bob 各自选择自己的私钥 $k_A, k_B < n$;
 - Alice 与 Bob 分别计算公钥 $P_A = k_A G, P_B = k_B G$, 并交换;
 - 计算密钥 $K = k_A P_B = k_B P_A$ 。
-
- 因为 $K = k_A k_B G$, 所以这两个密钥是一样的。
 - 由于椭圆曲线对数问题是单向函数, 所以 Eve 无法获知 K 。

举例：ECC Diffie-Hellman 密钥交换

例

- $E_p(0, -4)$, 即 $y^2 = x^3 - 4$, $G = (2, 2)$, $p = 211$, $n = 240$;
- 计算 $240G = O$;
- $k_A = 121$, $P_A = 121(2, 2) = (115, 48)$;
- $k_B = 203$, $P_B = 203(2, 2) = (130, 203)$;
- $K = 121(130, 203) = 203(115, 48) = (161, 69)$ 。

椭圆曲线加密：明文嵌入

- 将明文消息 m 编码为点 $P_m = (x, y)$, 即明文嵌入。
- 选择整数 κ , 通常 $30 \leq \kappa \leq 50$, 对消息 m 计算如下一系列 x
$$x = m\kappa + i \quad i = 0, 1, 2, \dots$$
直到 $\exists y$ 满足 $y^2 = x^3 + ax + b \pmod{p}$ 。
- 因为 $0 \sim p$ 的整数中有一半是模 p 平方剩余, 所以尝试了 r 次之后, 找到一个平方剩余的概率不小于 $1 - 2^{-r}$ 。

例 (明文嵌入)

- $y^2 = x^3 + 3x \pmod{4177}$, 消息 $m = 2174$, 取 $k = 30$
- 当 $i = 15$ 时, $x = mk + i = 65235$, $x^3 + 3x \pmod{p} = 38^2$
- 所以得到椭圆曲线上的点 $(65235, 38)$
- 若已知点 $(65235, 38)$, 则明文 $m = \lfloor \frac{65235}{k} \rfloor = \lfloor 2174.5 \rfloor = 2174$

椭圆曲线密码的安全性

- ECC 的安全性是建立在由 kP 和 P 确定 k 的难度之上的，即椭圆曲线对数问题。
- ECC 可以使用比 RSA 短得多的密钥。
- 密钥长度相同时，ECC 与 RSA 所执行的计算量也差不多。
- 与具有同等安全性的 RSA 相比，由于 ECC 使用的密钥更短，所以 ECC 所需的计算量比 RSA 少。

Symmetric	56	80	112	128	192	256
RSA n	512	1024	2048	3072	7680	15360
ECC p	112	161	224	256	384	512
Key size ratio	5:1	6:1	9:1	12:1	20:1	30:1

小结

- 1 数论基础
- 2 基本概念与 RSA 算法
- 3 Diffie-Hellman 密钥交换协议
- 4 椭圆曲线密码