第 10 章 离散对数

离散对数理论是现代公钥密码学的重要理论基础. 本章介绍离散对数的概念和性质, 并介绍几种常用的离散对数算法.

10.1 离散对数的概念及算法

定义 10.1.1 设 H 为有限群, $\alpha, \beta \in H$. 则定义**离散对数问题（DLP）**为: 求整数 x, 使得 $\beta = \alpha^x$, 此时称 x 为 H 上以 α 为底的 β 的**离散对数**, 记为

$$x = \log_\alpha \beta (\mathrm{mod}\ |H|).$$

显然, 如果 H 为有限循环群, α 是它的生成元, 则任意 $\beta \in H$, 以 α 为底的离散对数一定存在.

在密码学实践中, 最常用的离散对数有两大类：循环群 $\mathbf{Z}_p^*$ 上的离散对数和椭圆曲线上的离散对数, 我们先来介绍 $\mathbf{Z}_p^*$ 上的离散对数.

设 p 为素数, 如果 α 是 p 的原根, 则

$$\beta \equiv \alpha^x\ (\mathrm{mod}\ p)\ (0 \leqslant x \leqslant p-1) \tag{10.1.1}$$

遍历 $\mathbf{Z}_p^*$ 中的所有元素, 对任意的与 p 互素的整数 β, 我们可以找到唯一的整数 x 满足式(10.1.1). 此时 x 称为 $\mathbf{Z}_p^*$ 上以 α 为底的 β 的离散对数, 记为 $x = \log_\alpha \beta (\mathrm{mod}\ p-1)$.

例 10.1.1 已知 2 是素数 13 的一个原根, 则 $\mathbf{Z}_{13}^*$ 内关于 β = 1, 2, …, 12 的离散对数如表 10.1.1 所示

表 10.1.1 例 10.1.1 离散对数表

β	1	2	3	4	5	6	7	8	9	10	11	12
$\log_2\beta$ (mod 12)	12	1	4	2	9	5	11	3	8	10	7	6

定理 10.1.1 设 p 为素数, α 是 p 的原根, $\mathbf{Z}_p^*$ 上的离散对数具有如下性质：

(1) $\log_\alpha \alpha = 1 (\mathrm{mod}\ p\text{-}1)$,

(2) $\log_\alpha 1 = 0 (\mathrm{mod}\ p\text{-}1)$,

(3) 对任意与 p 互素的整数 b_1 和 b_2, $\log_\alpha b_1 b_2 \equiv \log_\alpha b_1 + \log_\alpha b_2\ (\mathrm{mod}\ p-1)$,

(4) 对任意整数 r 和任意 p 互素的整数 b, $\log_\alpha b^r \equiv r \log_\alpha b\ (\mathrm{mod}\ p-1)$.

这些性质可以很容易地利用初等数论的知识证明, 留作练习.

设 p 为素数, α 是 p 的原根, 已知 α, x, p 计算模指数 α^x (mod p)并不困难. 但是已知 β, 寻找 x 满足 $\beta \equiv \alpha^x$ (mod p), 却是一个计算量非常大的问题, 特别是当 p 是非常大的整数（例如 $p > 2^{1024}$）时. $\mathbf{Z}_p^*$ 上离散对数问题的这种性质称为**单向性**, 这种性质使得离散对数问题可以用来构造公钥密码学中所需要的单向函数. 下面介绍几种常用的离散对数求解算法:

1、穷举法（Brute-Force Search）

设 p 为素数, α 是 p 的原根, β 为任意与 p 互素的整数, 让 x 遍历 0 到 $p-1$ 中之间的每一个整数, 则一定能找到满足式(10.1.1)的 x, 这就是求解离散对数的穷举法. 这种算法在最坏的情况下需要做 p 次模幂运算, 平均需要 $p/2$ 次模幂运算, 其时间复杂度为 $O(p)$. 当 p 是非常大的整数时（例如 $p > 2^{1024}$）, 这种算法在计算上是不可行的.

2、商克法（Shanks' Baby-Step-Giant-Step Method）

商克法是基于“分治策略”来求解离散对数问题，其时间复杂度比穷举法要低，但需要额外的存储空间．商克法的基本思路如下：

取 $m=\left\lceil\sqrt{p-1}\right\rceil$，设法寻找整数 x_g, x_b，使得

$$x = x_g m + x_b \quad (0 \le x_g,\ x_b < m) \tag{10.1.2}$$

也就是寻找 $0 \le x_g, x_b < m$，使得 $\alpha^{x_g m + x_b} \equiv \beta \pmod{p}$，即

$$\beta(\alpha^{-m})^{x_g} \equiv \alpha^{x_b} \pmod{p} \tag{10.1.3}$$

只要找到了满足式(10.1.3)的整数 x_g 和 x_b，就可以代入式(10.1.2)求得 x.

运用分治策略寻找 x_g 和 x_b，分两步进行：

baby-step：对所有 x_b 的所有可能取值，计算 $\alpha^{x_b} \pmod{p}$（$0 \le x_b < m$），并按 $\alpha^{x_b} \pmod{p}$ 的升序排列存储序偶($\alpha^{x_b} \pmod{p}$, x_b)以备检索，于是得到一张包含 m 项的表，建立该表需要做 m 次模幂运算.并需要 m 个存储空间.

giant-step：对于 x_g 在范围 $0 \le x_g < m$ 内的所有可能取值，计算 $\beta(\alpha^{-m})^{x_g} \pmod{p}$ 并在 baby-step 中建立的表中查找是否存在某个 x_b 满足式(10.1.3)，如果存在，则此时的 x_g 和 x_b 即我们所求，代入(10.1.2)得 $x = x_g m + x_b$.

例 10.1.2 已知 $\alpha = 11$ 是 29 的一个生成元，在 $\mathbf{Z}_{29}^*$中求以α为底的 $\beta = 3$ 的离散对数.

解：$p = 29$，取 $m = \left\lceil\sqrt{p-1}\right\rceil = 6$.

baby-step：对所有 $x_b \in \{0, 1, 2, 3, 4, 5\}$计算 $\alpha^{x_b} \pmod{p}$，并按 $\alpha^{x_b} \pmod{p}$ 的升序排列存储($a^{x_b} \pmod{p}$, x_b)得下表：

x_b	0	2	1	5	4	3
$\alpha^{x_b} \equiv 11^{x_b} \pmod{29}$	1	5	11	14	25	26

giant-step：对于 x_g 在范围 $0 \le x_g < m$ 内的所有可能取值，计算 $\beta(\alpha^{-m})^{x_g} \pmod{p}$ 并在 baby-step 中建立的表中查找是否存在某个 x_b 满足式(10.1.3)，如果存在，则此时的 x_g 和 x_b 即我们所求，列表计算如下：

x_g	0	1	2
$\beta(\alpha^{-m})^{x_g} \equiv 3(11^{-6})^{x_g} \equiv 3(13)^{x_g} \pmod{29}$	3	10	14

所以 $x \equiv x_g m + x_b \equiv 2\times6+5 \equiv 17 \pmod{28}$.

商克法算法的时间复杂度为$O(\sqrt{p})$，空间复杂度也为$O(\sqrt{p})$，这是一个全指数级的复杂度．当 p 是非常大的整数时（例如 $p > 2^{1024}$），这种算法在计算上也是不可行的.

3、Pollard's ρ算法

Pollard's ρ算法是一种随机算法．其基本思路如下：

随机生成两个整数数列 a_i 和 b_i（$i = 0, 1, 2, \ldots$），并计算$\alpha^{a_i}\beta^{b_i} \pmod{p}$，直到找到 i_s, i_t，使得

$$\alpha^{a_{i_s}}\beta^{b_{i_s}} \equiv \alpha^{a_{i_t}}\beta^{b_{i_t}} \pmod{p} \tag{10.1.4}$$

此时

$$a_{i_s} + x b_{i_s} \equiv a_{i_t} + x b_{i_t} \pmod{p-1}.$$

于是有

$$x \equiv \frac{a_{i_t} - a_{i_s}}{b_{i_s} - b_{i_t}} \pmod{p-1}$$

例 10.1.3 已知α = 11 是循环群 $G = \mathrm{Z}_{29}^*$的一个生成元，求β = 3 的离散对数

解: 设 $x_i \equiv \alpha^{a_i}\beta^{b_i} \pmod{p}$, $a_0 = b_0 = 0$，$x_0 = \alpha^0\beta^0 = 1$，采用以下伪随机策略改变 a_i 和 b_i 的值:

- 如果 $x_{i-1} \pmod 3 = 0$, $a_i = 2a_{i-1} \pmod{|G|}$, $b_i = 2b_{i-1} \pmod{|G|}$
- 如果 $x_{i-1} \pmod 3 = 1$, $a_i = a_{i-1}+1 \pmod{|G|}$, $b_i = b_{i-1}$
- 如果 $x_{i-1} \pmod 3 = 2$, $a_i = a_{i-1}$, $b_i = b_{i-1}+1 \pmod{|G|}$

因为 G 为有限群，因此序列最终序列 x_i 必然满足周期性．即存在唯一的最小整数 $\mu \geq 0$ 以及 $\lambda \geq 1$，使得 $i > \mu$ 时，$x_{i+\lambda} = x_i$，而 $x_1, \ldots, x_{\mu+\lambda-1}$ 各不相同．整数称 μ 为序列 x_i 的前周期，而整数 λ 称为序列 x_i 的前周期

列表计算如下:

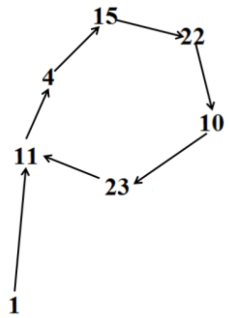
i	0	1	2	3	4	5	6	7
$\alpha^{a_i}\beta^{b_i}$	$11^0 3^0$	$11^1 3^0$	$11^1 3^1$	$11^2 3^1$	$11^4 3^2$	$11^5 3^2$	$11^6 3^2$	$11^6 3^3$
x_i	1	11	4	15	22	10	23	11

得到 $x_7 = x_1$，所以

$$x = \log_{11}3 \equiv (6-1)/(0-3) \pmod{28} \equiv 5(-3)^{-1} \pmod{28} \equiv 45 \pmod{28} \equiv 17 \pmod{28}.$$

注意，本题采用来改变 a_i 和 b_i 的值的伪随机策略，只是众多可用的伪随机策略之一，也可以采用其它伪随机策略来解此题.

这种算法只所以被称为 Pollard's ρ算法，是因为在计算过程中，每算出一个新的 x_i，都要“回头”看一看前面是否有重复的值，如果有重复的值，则算法停止．计算过程中的所有 x_i 可以连成一个希腊字母ρ的形状（也可能没有“尾巴”）.



可以证明 Pollard's ρ算法的时间复杂度也为 $O(\sqrt{p})$. 到目前为止，$\mathbf{Z}_p^*$ 上的离散对数问题的最好的求解的时间复杂度是亚指数级的，人们还没有找到复杂度在 $O((\log n)^k)$ (k 为常数) 内的算法．可见，当 p 非常大时，计算离散对数是非常困难的．正是利用离散对数的难解性，人们设计了各种基于离散对数的密码体制.

习题 10.1

A 组:

1、已知 $\alpha = 3$ 是循环群 Z_{17}^* 的一个生成元，分别用商克法和 Pollard's ρ法求 $\beta = 15$ 的离散对数.

2、已知 $\alpha = 5$ 是循环群 Z_{23}^* 的一个生成元，分别用商克法和 Pollard's ρ法求 $\beta = 10$ 的离散对数.

3、已知 $\alpha = 2$ 是循环群 Z_{37}^* 的一个生成元，分别用商克法和 Pollard's ρ法求 $\beta = 26$ 的离散对数.

B 组:

1、已知 $\alpha = 6$ 是循环群 Z_{26959}^* 的一个生成元，分别用商克法和 Pollard's ρ法求 $\beta = 25$ 的离散对数(编程计算)

10.2 椭圆曲线上的离散对数

椭圆曲线上的离散对数是循环群上的离散对数的一种特例. 给定有限域 $\mathbf{Z}_q$（$q = p^r$ 为素数幂)上的一条椭圆曲线 E, 并给定这条曲线上的两点 P 和 Q, 求正整数 k(如果存在的话）使之满足 $Q = kP$ 的问题, 称为椭圆曲线上的离散对数问题(ECDLP). 当点 P 的阶为大素数时, 普遍认为 ECDLP 是难解的. 反过来, 由第 9 章的知识, 已知 E 上的一点 P 和正整数 k, 求 E 上的另一点 $Q = kP$ 则是很容易的.

与一般的离散对数问题一样, 穷举法、商克法、Pollard's ρ算法也适用于椭圆曲线上的离散对数问题. 下面举一个例子:

例 10.2.1 研究椭圆曲线 E: $y^2 = x^3 + 8x + 8 \pmod{19}$上的离散对数问题, 已知$|E| = 13$, $P = (6, 14)$, $Q = (9, 7)$是 E 上的点, 用 Pollard's ρ算法求正整数 k 使之满足 $Q = kP$.

解 设$T_i \equiv a_iP + b_iQ$, 其中 a_i, b_i 为整数, $a_0 = b_0 = 1$，$T_0 = P+Q = (1, 6)$, 令 x_{T_i} 表示 T_i 的横坐标. 采用以下伪随机策略改变 a_i 和 b_i 的值:

- 如果 $x_{T_{i-1}} \pmod 3 = 0$，$a_i = 2a_{i-1} \pmod{|E|}$ ，$b_i= 2b_{i-1} \pmod{|E|}$
- 如果 $x_{T_{i-1}} \pmod 3 = 1$，$a_i = a_{i-1}+1 \pmod{|E|}$ ，$b_i= b_{i-1}$
- 如果 $x_{T_{i-1}} \pmod 3 = 2$，$a_i = a_{i-1}$，$b_i= b_{i-1}+1 \pmod{|E|}$
- 列表计算如下:

i	0	1	2	3	4
$a_iP + b_iQ$	$P+Q$	$2P+Q$	$3P+Q$	$4P+Q$	$8P+2Q$
T_i	(1, 6)	(10, 10)	(4,3)	(6,5)	(4,3)

所以, $3P+Q = 8P+2Q$, $5P + Q = O$, $Q = -5P = 8P$, $k = 8$.

习题 10.1

A 组:

1、研究椭圆曲线 E: $y^2 = x^3 + 4x + 4 \pmod{11}$上的离散对数问题, 已知$|E| = 11$, $P = (2, 8)$, $Q = (8, 3)$是 E 上的点, 用穷举法和 Pollard's ρ算法求正整数 k 使之满足 $Q = kP$.

2、研究椭圆曲线 E: $y^2 = x^3 + 4x + 2 \pmod{13}$上的离散对数问题, 已知$|E| = 13$, $P = (4, 11)$, $Q = (7, 3)$是 E 上的点, 用穷举法和 Pollard's ρ算法求正整数 k 使之满足 $Q = kP$.

B 组:

1、研究椭圆曲线 E: $y^2 = x^3 + 4x + 13 \pmod{83}$上的离散对数问题, 已知$|E| = 73$, $P = (19, 68)$, $Q = (74, 24)$是 E 上的点, 用穷举法和 Pollard's ρ算法求正整数 k 使之满足 $Q = kP$.

附录 1 10000 以内的素数表

23 5 7 11 13 17 19 23 2931 37 41 43 47 53 59 61 67 7173 79 83 89 97 101 103 107 109 113127
131 137 139 149 151 157 163 167 173179 181 191 193 197 199 211 223 227 229233 239 241 251
257 263 269 271 277 281283 293 307 311 313 317 331 337 347 349353 359 367 373 379 383 389
397 401 409419 421 431 433 439 443 449 457 461 463467 479 487 491 499 503 509 521 523
541547 557 563 569 571 577 587 593 599 601607 613 617 619 631 641 643 647 653 659661 673
677 683 691 701 709 719 727 733739 743 751 757 761 769 773 787 797 809811 821 823 827 829
839 853 857 859 863877 881 883 887 907 911 919 929 937 941947 953 967 971 977 983 991 997
1009 1013 1019 1021 1031 1033 1039 1049 1051 10611063 1069 1087 1091 1093 1097 1103 1109
1117 11231129 1151 1153 1163 1171 1181 1187 1193 1201 12131217 1223 1229 1231 1237 1249
1259 1277 1279 12831289 1291 1297 1301 1303 1307 1319 1321 1327 13611367 1373 1381 1399
1409 1423 1427 1429 1433 14391447 1451 1453 1459 1471 1481 1483 1487 1489 14931499 1511
1523 1531 1543 1549 1553 1559 1567 15711579 1583 1597 1601 1607 1609 1613 1619 1621
16271637 1657 1663 1667 1669 1693 1697 1699 1709 17211723 1733 1741 1747 1753 1759 1777
1783 1787 17891801 1811 1823 1831 1847 1861 1867 1871 1873 18771879 1889 1901 1907 1913
1931 1933 1949 1951 19731979 1987 1993 1997 1999 2003 2011 2017 2027 20292039 2053 2063
2069 2081 2083 2087 2089 2099 21112113 2129 2131 2137 2141 2143 2153 2161 2179 22032207
2213 2221 2237 2239 2243 2251 2267 2269 22732281 2287 2293 2297 2309 2311 2333 2339 2341
23472351 2357 2371 2377 2381 2383 2389 2393 2399 24112417 2423 2437 2441 2447 2459 2467
2473 2477 25032521 2531 2539 2543 2549 2551 2557 2579 2591 25932609 2617 2621 2633 2647
2657 2659 2663 2671 26772683 2687 2689 2693 2699 2707 2711 2713 2719 27292731 2741 2749
2753 2767 2777 2789 2791 2797 28012803 2819 2833 2837 2843 2851 2857 2861 2879 28872897
2903 2909 2917 2927 2939 2953 2957 2963 29692971 2999 3001 3011 3019 3023 3037 3041 3049
30613067 3079 3083 3089 3109 3119 3121 3137 3163 31673169 3181 3187 3191 3203 3209 3217
3221 3229 32513253 3257 3259 3271 3299 3301 3307 3313 3319 33233329 3331 3343 3347 3359
3361 3371 3373 3389 33913407 3413 3433 3449 3457 3461 3463 3467 3469 34913499 3511 3517
3527 3529 3533 3539 3541 3547 35573559 3571 3581 3583 3593 3607 3613 3617 3623 36313637
3643 3659 3671 3673 3677 3691 3697 3701 37093719 3727 3733 3739 3761 3767 3769 3779 3793
37973803 3821 3823 3833 3847 3851 3853 3863 3877 38813889 3907 3911 3917 3919 3923 3929
3931 3943 39473967 3989 4001 4003 4007 4013 4019 4021 4027 40494051 4057 4073 4079 4091
4093 4099 4111 4127 41294133 4139 4153 4157 4159 4177 4201 4211 4217 42194229 4231 4241
4243 4253 4259 4261 4271 4273 42834289 4297 4327 4337 4339 4349 4357 4363 4373 43914397
4409 4421 4423 4441 4447 4451 4457 4463 44814483 4493 4507 4513 4517 4519 4523 4547 4549
45614567 4583 4591 4597 4603 4621 4637 4639 4643 46494651 4657 4663 4673 4679 4691 4703
4721 4723 47294733 4751 4759 4783 4787 4789 4793 4799 4801 48134817 4831 4861 4871 4877
4889 4903 4909 4919 49314933 4937 4943 4951 4957 4967 4969 4973 4987 49934999 5003 5009
5011 5021 5023 5039 5051 5059 50775081 5087 5099 5101 5107 5113 5119 5147 5153 51675171
5179 5189 5197 5209 5227 5231 5233 5237 52615273 5279 5281 5297 5303 5309 5323 5333 5347
53515381 5387 5393 5399 5407 5413 5417 5419 5431 54375441 5443 5449 5471 5477 5479 5483
5501 5503 55075519 5521 5527 5531 5557 5563 5569 5573 5581 55915623 5639 5641 5647 5651
5653 5657 5659 5669 56835689 5693 5701 5711 5717 5737 5741 5743 5749 57795783 5791 5801
5807 5813 5821 5827 5839 5843 58495851 5857 5861 5867 5869 5879 5881 5897 5903 59235927
5939 5953 5981 5987 6007 6011 6029 6037 60436047 6053 6067 6073 6079 6089 6091 6101 6113

61216131 6133 6143 6151 6163 6173 6197 6199 6203 62116217 6221 6229 6247 6257 6263 6269 6271 6277 62876299 6301 6311 6317 6323 6329 6337 6343 6353 63596361 6367 6373 6379 6389 6397 6421 6427 6449 64516469 6473 6481 6491 6521 6529 6547 6551 6553 65636569 6571 6577 6581 6599 6607 6619 6637 6653 66596661 6673 6679 6689 6691 6701 6703 6709 6719 67336737 6761 6763 6779 6781 6791 6793 6803 6823 68276829 6833 6841 6857 6863 6869 6871 6883 6899 69076911 6917 6947 6949 6959 6961 6967 6971 6977 69836991 6997 7001 7013 7019 7027 7039 7043 7057 70697079 7103 7109 7121 7127 7129 7151 7159 7177 71877193 7207 7211 7213 7219 7229 7237 7243 7247 72537283 7297 7307 7309 7321 7331 7333 7349 7351 73697393 7411 7417 7433 7451 7457 7459 7477 7481 74877489 7499 7507 7517 7523 7529 7537 7541 7547 75497559 7561 7573 7577 7583 7589 7591 7603 7607 76217639 7643 7649 7669 7673 7681 7687 7691 7699 77037717 7723 7727 7741 7753 7757 7759 7789 7793 78177823 7829 7841 7853 7867 7873 7877 7879 7883 79017907 7919 7927 7933 7937 7949 7951 7963 7993 80098011 8017 8039 8053 8059 8069 8081 8087 8089 80938101 8111 8117 8123 8147 8161 8167 8171 8179 81918209 8219 8221 8231 8233 8237 8243 8263 8269 82738287 8291 8293 8297 8311 8317 8329 8353 8363 83698377 8387 8389 8419 8423 8429 8431 8443 8447 84618467 8501 8513 8521 8527 8537 8539 8543 8563 85738581 8597 8599 8609 8623 8627 8629 8641 8647 86638669 8677 8681 8689 8693 8699 8707 8713 8719 87318737 8741 8747 8753 8761 8779 8783 8803 8807 88198821 8831 8837 8839 8849 8861 8863 8867 8887 88938923 8929 8933 8941 8951 8963 8969 8971 8999 90019007 9011 9013 9029 9041 9043 9049 9059 9067 90919103 9109 9127 9133 9137 9151 9157 9161 9173 91819187 9199 9203 9209 9221 9227 9239 9241 9257 92779281 9283 9293 9311 9319 9323 9337 9341 9343 93499371 9377 9391 9397 9403 9413 9419 9421 9431 94339437 9439 9461 9463 9467 9473 9479 9491 9497 95119521 9533 9539 9547 9551 9587 9601 9613 9619 96239629 9631 9643 9649 9661 9677 9679 9689 9697 97199721 9733 9739 9743 9749 9767 9769 9781 9787 97919803 9811 9817 9829 9833 9839 9851 9857 9859 98719883 9887 9901 9907 9923 9929 9931 9941 9949 99679973

附录 2　200 以内素数的最小原根表

p	g	p	g	p	g	p	g	p	g	p	g
2	1	23	5	59	2	97	5	137	3	179	2
3	2	29	2	61	2	101	2	139	2	181	2
5	2	31	3	67	2	103	5	149	2	191	19
7	3	37	2	71	7	107	2	151	6	193	5
11	2	41	6	73	5	109	6	157	5	197	2
13	2	43	3	79	3	113	3	163	2	199	3
17	3	47	5	83	2	127	3	167	5		
19	2	53	2	89	3	131	2	173	2		