

# 实验二 离散对数求解

## 实验目的

- 掌握与密码学相关的基础数论知识；
- 利用中间相遇攻击来编程实现离散对数的求解。

## 实验内容

- 计算模素数 $p$ 的离散对数
- 令 $p$ 是一个素数， $g$ 是有限乘法群 $Z_p^*$ 上的一个原根，然后给定一个 $Z_p^*$ 上的 $h$ 满足 $h = g^x$ ，其中 $1 \leq x \leq 240$ ，目的是找到 $x$ 。也就是说，编写的程序以 $p, g, h$ 作为输入，然后输出 $x$ 。
- 该问题最直接的算法就是对 $x$ 的 $2^{40}$ 个可能的值逐个进行尝试，直到找到正确的一个，即直到找到一个 $x$ 在 $Z_p$ 上满足 $h = g^x$ 。这需要 $2^{40}$ 次乘法运算。在本次实验中，需要实现一个算法，该算法使用中间相遇攻击，时间代价约为 $\sqrt{2^{40}} = 2^{20}$ 。
- 令 $B = 2^{20}$ 。因为 $x$ 是小于 $B^2$ ，我们可以将未知的 $x$ 写作 $x = x_0 B + x_1$ ，其中 $x_0, x_1 \in [0, B - 1]$ 。然后， $h = g^x = g^{x_0 B + x_1} = (g^B)^{x_0} \times g^{x_1}$ （在 $Z_p$ 上）。两边同时除以 $g^{x_1}$ ，可得到 $h/g^{x_1} = (g^B)^{x_0}$ （在 $Z_p$ 上）。
- 上面等式中的变量是 $x_0$ 和 $x_1$ ，其他都是已知的： $g$ 和 $h$ 是给定的， $b = 2^{20}$ 。由于 $x_0$ 和 $x_1$ 在等式的两边，所以我们可以使用中间相遇攻击来找到一个解：
  - 为等式左边 $h/g^{x_1}$ 的所有可能值创建一个哈希表，其中 $x_1 = 0, 1, \dots, 2^{20}$ 。
  - 对于每一个 $x_0 = 0, 1, \dots, 2^{20}$ ，检查 $(g^B)^{x_0}$ 是否在哈希表中，如果是，便找了解 $(x_0, x_1)$ ，即 $x = x_0 B + x_1$ 。
  - 总体工作大约是 $2^{20}$ 次乘法来构建表，另外 $2^{20}$ 次查找在此表中。
  - 当完成求解程序之后，请以附件test.txt中的 $p, g, h$ 为输入，求解出 $x$ 。
- **提示：本次实验需要使用支持多精度和模运算的环境。在Python中，可以使用gmpy2或numbthy模块。在C/C++中，可以使用GMP。**

## 实验时间地点

- 4月9日（周五）晚上
- 思贤楼303机房

## 编程语言

- Python或者 C/C++或者Java

## 实验要求

- 在线提交源码和实验报告；
- 实验报告需包括实验结果（ $x_0, x_1, x$ 的值）、重要代码段解释以及本次实验总结；
- 实验代码禁止抄袭，可以在网上进行参考，但是如果没有任何改变，将判定抄袭，本次实验记0分；
- 鼓励大家思考新的求解方法和代码，对于能够用新思路求解出较好结果的同学本次实验加分的奖励；

- 实验报告截止时间为4月16日。

本次实验设立3个java小助教，2个C++小助教，4个python小助教，欢迎大家提前来找我们检查。