《现代密码学》实验报告

实验名称：PRG	实验时间：2022.12.27
学生姓名：伍建霖	学号：20337251
学生班级：20网安	成绩评定：

二、实验内容

使用C++语言，实现伪随机数生成器。

三、实验原理

设p是一个64比特长的素数，$g \in \mathbb{Z}_p^*$是一个本原元。种子s_0是$\mathbb{Z}_p^*$中的一个元素。

对于$0 \leq i \leq l-1$，定义

$$s_{i+1} = g^{s_i} \mod p$$

然后定义

$$f(s_0) = (z_1, z_2, \ldots, z_l)$$

其中

$$z_i = s_i \mod 2 (1 \leq i \leq l)$$

最终程序输出$f(s_0) = (z_1, z_2, \ldots, z_l)$。

四、实验步骤

modulo

模运算助教已经给出了伪代码，从伪代码上看只需要实现比较，移位和减法操作。比较操作通过 if 语句可以实现；移位操作通过 uint64_t 里的移位也可以实现；减法操作通过 uint64_t 里的减法也可以实现，需要判断是否从高64位借位。

```
X = B;

while (X <= A/2)
{
    X <<= 1;
}

while (A >= B)
{
    if (A >= X)
        A -= X;
    X >>= 1;
}

return A;
```

multiply

两个64位的整数相乘，低64位直接获得，高64位就通过[c++ - Getting the high part of 64 bit integer multiplication - Stack Overflow]里的算法获得。

powm

有了模运算和乘法运算之后，模乘就好实现了。我们要计算 g^s ，而 s 又可以分解成 多个2的幂相加(忘记术语了)，比如 7 = 4+2+1，13 = 8+4+1这样，则有 $g^7 = g^4 * g^2 * g^1$ 这种形式，这样就很直观了。

然后为了提高性能，参考了之前有限域里看到的快速幂算法的思想。

五、实验结果

```
10000101101100100000111111111111011001110010010101100000000010101011011101110000101100101011011110110000001
00001111011010011000010010100011111111101101100001111000010110110101010101001100100111001111010001010110l
0011110101111101000110111101001001110001011101101011101110110001110100111000010111010101101111011110001l0
00011011101011111001110011110001110010111100011010111100000101000011001010101011111100011011001100100110l
0111000111001100111100011101001011001101010011000001000110110111100011010001001011010111101l
0: 236
1: 276

Time: 426.26
(base) PS D:\CodeField\Cryptography\lab>
```

六、实验总结

阳了，没精力优化和改原理了。

这次的实验结合助教给的提示并不算难，就是时间比较紧张，一开始的时候脑袋转不过来，想不出模乘的实现方法。