

山东大学
网络空间安全学院

密码工程第二次实验
实验报告

学生姓名 陈嘉衡-202200460178

胡承旭-202200460148

冯奕楠-202200460037

指导教师 王伟嘉

学 院 网络空间安全学院

专业班级 22 级密码一班

完成时间 2024 年 10 月 14 日

一、实验目标

使用任意两个 2048 位数字来实现蒙哥马利乘法和 Barrett 归约。

二、实验背景与原理

蒙哥马利算法和 Barrett 归约原理都是用于大数模运算的优化算法，尤其在公钥密码学中有着广泛的应用。以下是对这两种算法原理的详细解释：

蒙哥马利算法 (Montgomery Algorithm) 是一种用于快速模运算的算法，其主要思想就是简化除法运算，转化成位运算。

它主要分为蒙哥马利模乘、蒙哥马利约简和蒙哥马利模幂三种形式。

1. 蒙哥马利形式：

为了计算 $ab \pmod N$ ，找一个 R ，然后使得 $a' \equiv aR \pmod N, b' \equiv bR \pmod N$ ，

其中 R 需要满足： $R = 2^k > N$ (其中 k 是满足条件的最小正整数)， $\gcd(R, N) = 1$

2. 蒙哥马利约简 Montgomery reduction (a, b, N)：

计算 $XR^{-1} \pmod N$ ，由此前的 R 的定义可知， $R = 2^k$ ，所以 $XR^{-1} = X \gg k$ (X 右移 k 位)
但是右移 k 位可能会抹掉 X 的低位中的一些 1，这不是精确计算，而是想下取整的除法，
当且仅当 X 是 R 的整数倍时， X/R 严格等于 $X \gg k$ ，所以找到一个 m ，使得 $X+mR$ 是 R 的倍数，
计算这个数 $\pmod N$ 即可。

根据 R 的定义， $\gcd(R, N) = 1$ ，根据扩展的欧几里得算法，有 $RR' - NN' = 1$ 并且有 $0 < N' < R, 0 < R' < N < R$ 。(注意这里是 $-NN'$ ，所以有 $N' = -N^{-1} \pmod R$)

$$\begin{aligned} X + mN &\equiv 0 \pmod R \\ XN' + mNN' &\equiv 0 \pmod R \\ XN' + m(RR' - 1) &\equiv 0 \pmod R \\ XN' &\equiv m \pmod R \end{aligned}$$

约简流程如下：

计算 $N' = -N^{-1} \pmod R$ ，计算 $m = XN' \pmod R$ ；

计算 $y = \frac{X + mN}{R}$ ：将 $X + mN$ 右移 k 位；

若 $y > N$ ，则 $y = y - N$ ，这时的 y 满足： $0 < y < 2N$ 。

因为 $X < N^2, m < R, N < R$ ，所以 $\frac{X + mN}{R} < \frac{N^2 + RN}{R} < \frac{RN + RN}{R} = 2N$ ；

返回 y 。

3. 蒙哥马利模乘 Montgomery Multiply (a, b, N)：

计算 $a' \equiv aR \pmod N, b' \equiv bR \pmod N, X = a'b'$ ；

调用蒙哥马利约简， $X_1 = \text{Montgomeryreduction}(X, R, N) = X/R = abR \pmod R$ ；

再调用蒙哥马利约简, $y = \text{Montgomeryreduction}(X_1, R, N) = X_1 / R = ab(\text{mod } N)$;
返回 y 。

3. 蒙哥马利模幂:

是基于蒙哥马利模乘和蒙哥马利约简实现的幂运算。

Barrett 归约原理是一种用于大数模运算的优化方法, 它利用预计算和简单的算术运算来避免大整数除法。

1. 预计算:

在进行模运算之前, 预计算出一些常数, 以便在后续的归约过程中使用。

2. 归约过程:

通过乘法和加法运算, 将大整数转化为一个较小的范围, 然后利用预计算的常数进行调整, 最终得到模运算的结果。归约过程中避免了除法运算, 从而提高了计算效率。

三、实验环境与设置

(一) 硬件配置详情

处理器 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz

机带 RAM 16.0 GB (15.7 GB 可用)

系统类型 64 位操作系统, 基于 x64 的处理器

(二) 软件配置详情

操作系统版本: windows 11 家庭中文版 22H2

VirtualBox 版本 7.1.0 r164728 (Qt6.5.3)

编程语言: python 3.12

四、实验步骤与结果展示

实验代码:

```
import random

from Crypto.Util.number import getPrime, inverse

# 蒙哥马利约简
def montgomery_reduce(a, n, n_inv, r):
    # 计算中间变量 q
    q = ((a & (r - 1)) * n_inv) & (r - 1)
    # 计算约简后的 a
    a = (a + q * n) >> (r.bit_length() - 1)
    # 如果 a 大于等于模数 n, 则减去 n
    if a >= n:
```

```

        a -= n
    return a

# 蒙哥马利乘法
def montgomery_multiply(a_bar, b_bar, n, n_inv, r):
    # 计算乘积
    t = a_bar * b_bar
    # 对乘积进行蒙哥马利约简
    return montgomery_reduce(t, n, n_inv, r)

# 巴雷特约简
def barrett_reduction(x, n):
    # 计算模数 n 的位数
    k = n.bit_length()
    # 计算 2 的 2k 次幂
    r = 1 << (k * 2)
    # 计算 n 的逆元 (使用 Crypto 库的 inverse 函数)
    n_inv = inverse(n, r)
    # 计算中间变量 q2
    q2 = ((x * n_inv) >> k) >> k
    # 计算约简后的结果
    r = x - q2 * n
    # 如果结果大于等于 n, 则减去 n
    if r >= n:
        r -= n
    return r

# 生成两个随机的 2048 位数和一个 2048 位的素数模数
a = random.randint(2**2047, 2**2048 - 1)
b = random.randint(2**2047, 2**2048 - 1)
n = getPrime(2048)

# 初始化蒙哥马利乘法的参数
r = 1 << n.bit_length() # r 是大于 n 的最小 2 的幂
n_inv = inverse(n, r) # n 的模逆

```

```

# 将 a 和 b 转换为蒙哥马利形式
a_bar = (a * r) % n
b_bar = (b * r) % n

# 执行蒙哥马利乘法
montgomery_result = montgomery_multiply(a_bar, b_bar, n, n_inv, r)

# 将结果转换回普通形式
# 但为了与直接计算的结果对比，我们可以再次约简以确保一致性（实际上这一步是多余的）
montgomery_result_normal = montgomery_reduce(montgomery_result * n_inv, n, n_inv, r) % n

# 执行巴雷特约简
# 为了对比，我们直接对 a*b 的结果进行巴雷特约简
barrett_result = barrett_reduction(a * b, n)

# 输出结果
print("Answer")
print((a * b) % n)
print("Montgomery Multiplication (Normal Form)")
print(montgomery_result_normal)
print("Barrett Reduction ")
print(barrett_result)

```

实验结果:

```

D:\python\python.exe D:\qq文件\hw1.py
Answer
48519569141439529164103978862351913879056231490883995824496365771802678502204334973594370585631863146614631435355624613701457571561527953193788021147547507945697641954542732630141742007192103
52054477084369891173352350388645229240600410351709997672220478686038495370893820587140965662987827035579093718977234766191447057502379665650198068198743243122430987255201767442354231733357059
310654299742157079909388938494600092023074227751179856200668544078873357233316822317099040066075695207253911157710882083011757768835120187509643769746717172124632685309957620935681167692397918
97746318018154455048789013808848588355
Montgomery Multiplication (Normal Form)
125296850422372649772733098273687659489109890128529726114668634021711232103138752639005372258443580853734129434232365669684955552880817110098481182496475545164103083883813613927373084809271237
93009139760949824732189546370853456910266009260682662875958522020595252788624818415638680026557209721664942119148808441477968249227309068369957139153415014269800790103445036680561159466690070
915815101785265873969134457175481287245491960213349500292300252941795736955223898115878028750960008107828476212071313363982945084318946246884787276939779387487166282961632933153132020795193799
8040260799177173354805937192124809242604
Barrett Reduction
-87546275860929573573066153380418859454945906511779174977283751244741835316119990098776897596768942093640238559293175803215175050045821789276802203047088173411035646372612690644401308349304959
69285342290244793833260064275513429483256372542254666842038545768103981128710464120219880429668243384226041474395635138389243009904438314751723252351389182749547089405407008334713549377007309
09822861440784783239812422868562032791778684844616903559257708775826255271240255119665533566438435486603464950596850121944822473477437313551344996097881954131825133396819884162721167550722569
581021788074300382365606352005289152071888604485811174713697388284064603759441901131117614863039113373419771202448578008673876800047594562493543609587701598884358956730503998682939897325719
475766798748582932943961844830314514819388921904103135451809544827261149337163967060370438898829483932308954375478353923789092235836687419373677378135879025841497663630116156372562361009928312
64083731573064970786508007421631472527701255997085814437711380423078387084450015544342338987285443464593114039640404383830967636312151070784396650187865099937883273391384131330634064414534587
770800310077404432946023812864462708940979594635583780289438944380108978350418096371985002477827225492215782731895178866026951834109464083549530795854063423497528429605284017690177226292040926
8625115989310015381575066508897881354714333180266686169480427040526711350950709262398257764862770038375641878137349932585930688008761167450571579398486260079407736854797821029681609182780170
067722852901899339640784155506436744698467528071963887191709146360514977155283590784008783658291972513684914379828563105433391124285248239473865074124463072447603494555749090217584500415753965
9842839319516176367539845504166082521820367131894719180077907160099595651270871372018172517236567452559104953398595057841

```

