现代密码学作业——第六讲

1、由 n = 35,知 p=7，q=5.所以 Φ（35）=24，易知 d = 5，因此 M = 105 mod 35 = 5

2、

（1）C=59,57

（2）K=3 m=30

（3）K=2 m=19

3、 Miller-Rabin(n,t)

输入：一个大于 3 的奇整数 n 和一个大于等于 1 的安全参数 t(用于确定测试轮数)。

输出：返回 n 是否是素数(概率意义上的，一般误判概率小于(1/2)80 即可）。

1、将 n-1 表示成 2sr，(其 中 r 是奇数)

2、 对 i 从 1 到 t 循环作下面的操作：

2.1 选择一个随机整数 a(2≤a ≤n-2)

2.2 计算 y ←ar mod n

2.3 如果 y≠1 并且 y ≠n-1 作下面的操作,否则转 3：

2.3.1 j←1;

2.3.2 当 j≤s-1 并且 y≠n-1 循环作下面操作,否则跳

2.3.3：{计算 y ←y2 mod n;

如果 y=1 返回 合数 ；

否则 j←j+1; }

2.3.3 如果 y ≠n-1 则返回 合数 ；

3、返回素数。、

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <math.h>
using namespace std;

// 生成伪素数
const int MAX_ROW = 50;
size_t Pseudoprime()
{
    bool ifprime = false;
    size_t a = 0;
    int arr[MAX_ROW];   //数组 arr 为{3，4，5，6...52}
    for (int i = 0; i<MAX_ROW; ++i)
    {
        arr[i] = i+3;
    }
    while (!ifprime)
    {
        srand((unsigned)time(0));
        ifprime = true;
        a = (rand()%10000)*2+3; //生成一个范围在 3 到 2003 里的奇数
        for (int j = 0; j<MAX_ROW; ++j)
        {
            if (a%arr[j] == 0)
            {
                ifprime = false;
                break;
            }
        }
    }
    return a;
}

size_t  repeatMod(size_t base, size_t n, size_t mod)//模重复平方算法求(b^n)%m
{
    size_t a = 1;
    while(n)
    {
```

```
40.         if(n&1)
41.         {
42.             a = (a*base)%mod;
43.         }
44.         base = (base*base)%mod;
45.         n = n>>1;
46.     }
47.     return a;
48. }
49.
50. //Miller-Rabin 素数检测
51. bool rabinmiller(size_t n, size_t k)
52. {
53.
54.     int s = 0;
55.     int temp = n-1;
56.     while ((temp & 0x1) == 0 && temp)
57.     {
58.         temp = temp>>1;
59.         s++;
60.     }   //将 n-1 表示为(2^s)*t
61.     size_t t = temp;
62.
63.     while(k--)  //判断 k 轮误判概率不大于(1/4)^k
64.     {
65.         srand((unsigned)time(0));
66.         size_t b = rand()%(n-2)+2; //生成一个 b(2≤a ≤n-2)
67.
68.         size_t y = repeatMod(b,t,n);
69.         if (y == 1 || y == (n-1))
70.             return true;
71.         for(int j = 1; j<=(s-1) && y != (n-1); ++j)
72.         {
73.             y = repeatMod(y,2,n);
74.             if (y == 1)
75.                 return false;
76.         }
77.         if ( y != (n-1))
78.             return false;
79.     }
80.     return true;
81. }
```

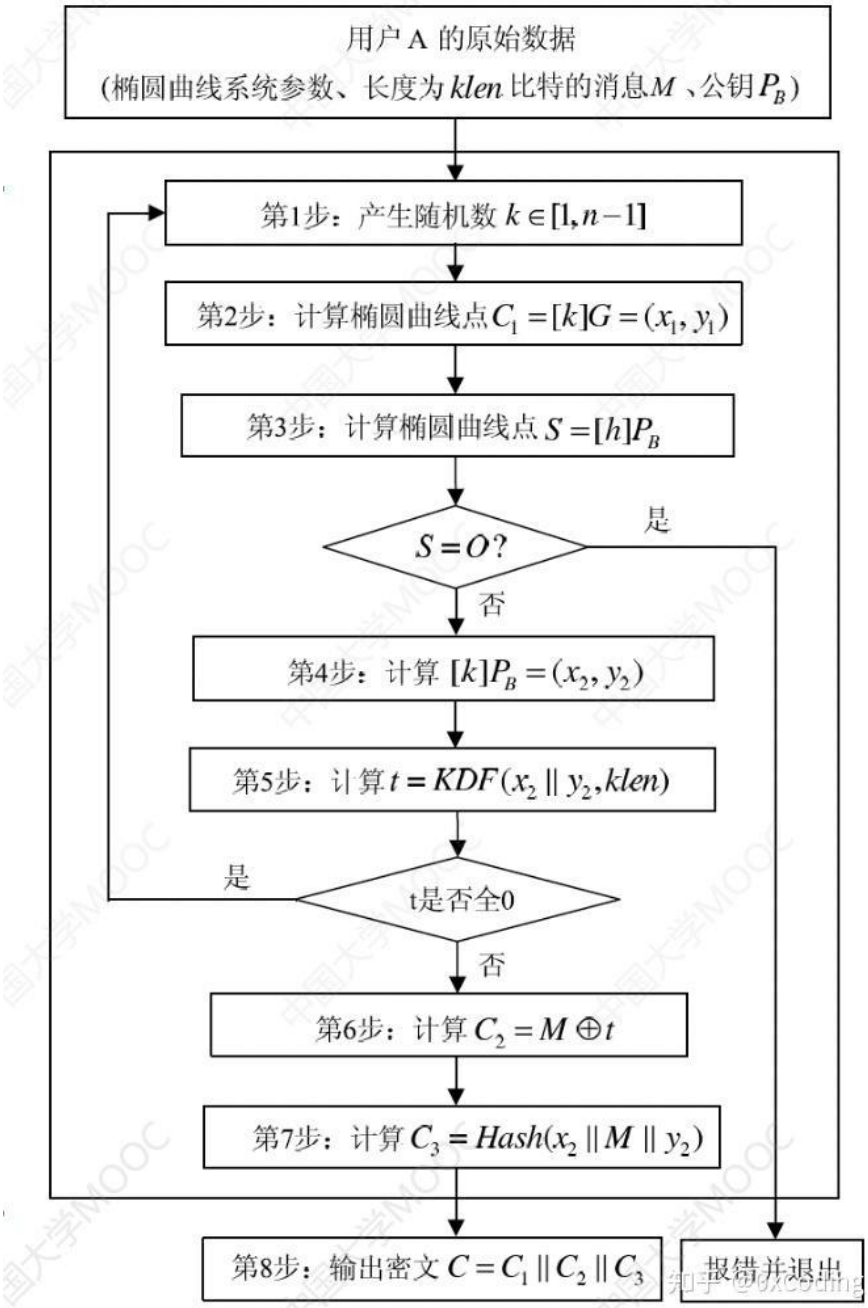
4、

密钥产生

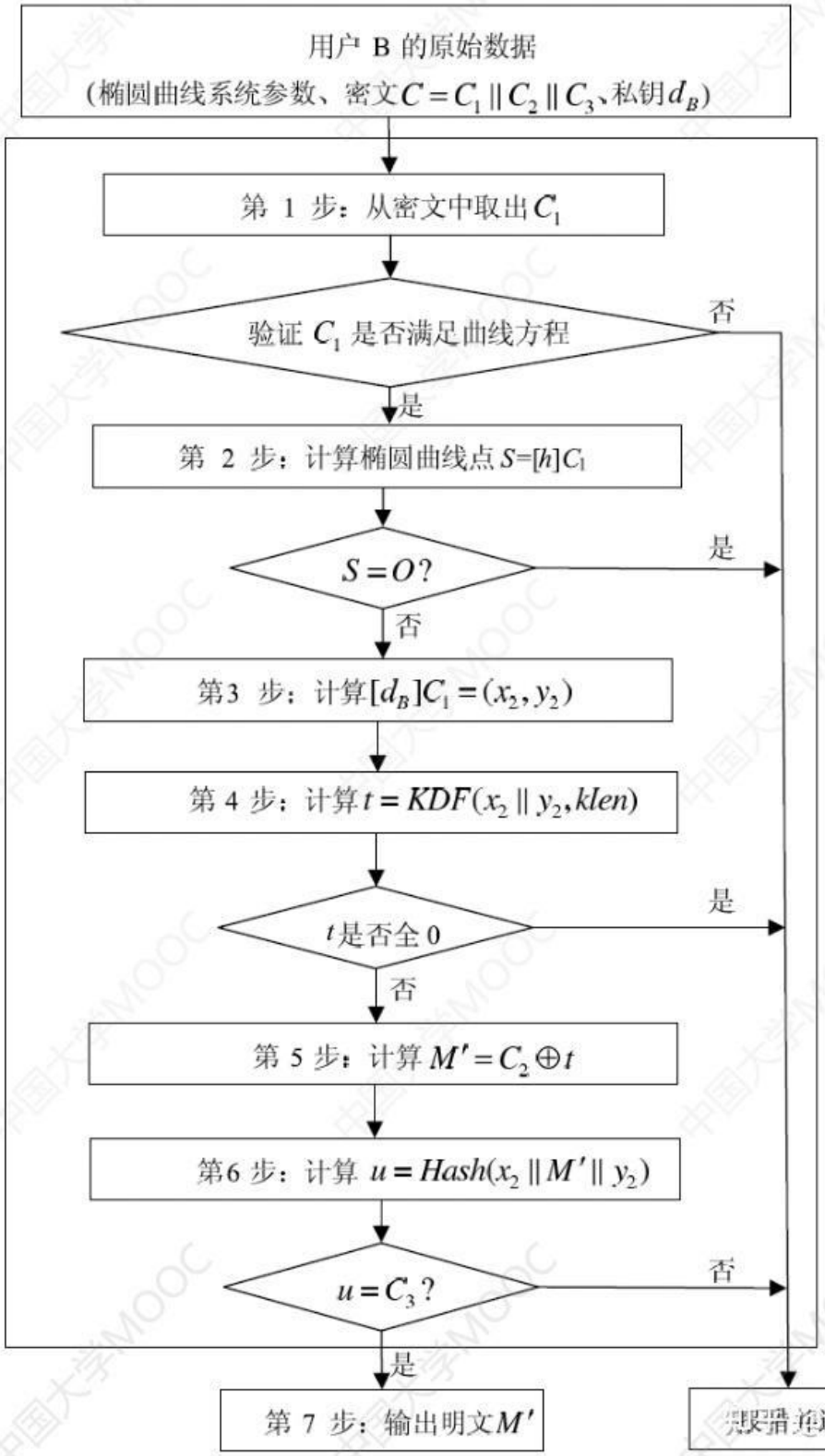
设接收方为B，B的密钥取为$\{1,2,...,n-1\}$的一个随机数dB，记为$dB \leftarrow R\{1,2,...,n-1\}$，其中$n$是基点$G$的阶。

B的公钥取为椭圆曲线上的点$PB=dBG$，其中$G=G(x,y)$是基点。

加密算法



解密算法


用户 B 的原始数据
(椭圆曲线系统参数、密文$C=C_1\|C_2\|C_3$、私钥d_B)
第 1 步：从密文中取出C_1
验证 C_1 是否满足曲线方程
否
是
第 2 步：计算椭圆曲线点$S=[h]C_1$
$S=O$?
是
否
第3 步：计算$[d_B]C_1=(x_2,y_2)$
第 4 步：计算$t=KDF(x_2\|y_2,klen)$
t是否全 0
是
否
第 5 步：计算$M'=C_2\oplus t$
第6 步：计算 $u=Hash(x_2\|M'\|y_2)$
$u=C_3$?
否
是
第 7 步：输出明文M'