《信息安全基础综合实验》课程实验报告

实验题目：基于中国剩余定理的秘密共享方案

班级：1718039　　学号 1：17189110002　　姓名 1：祝佳磊

班级：1718039　　学号 2：17180210027　　姓名 2：李欣

一、实验目的

实验环境:

1.Visual Studio2017

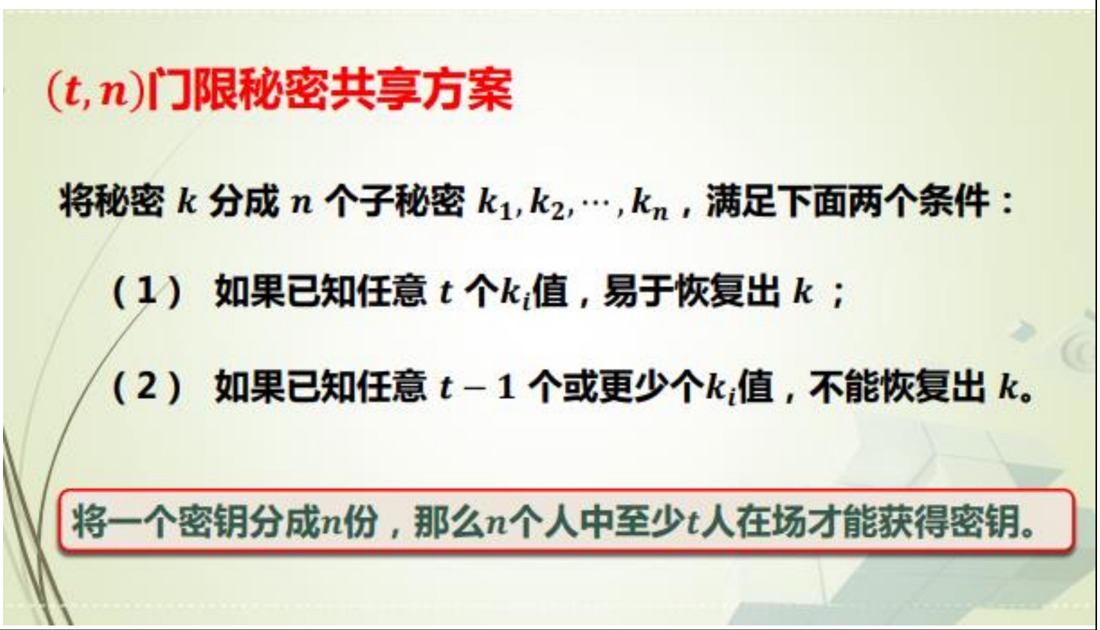
2.miracl 库

实现目标：使用中国剩余定理完成秘密共享

二、方案设计

（包括背景、原理、必要的公式、图表、算法步骤等等）

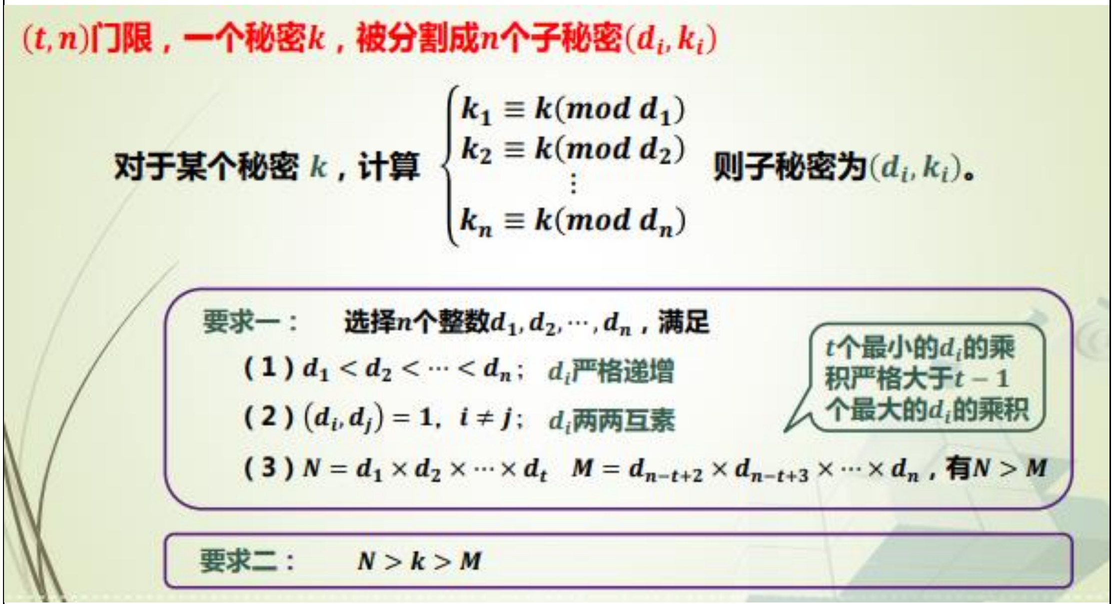
背景：秘密共享是将秘密以适当的方式拆分，拆分后的每一个子秘密由不同的参与者管理，单个参与者无法恢复秘密信息，只有若干个参与者一同协作才能恢复秘密消息。并且，当其中某些参与者出问题时，秘密仍可以恢复

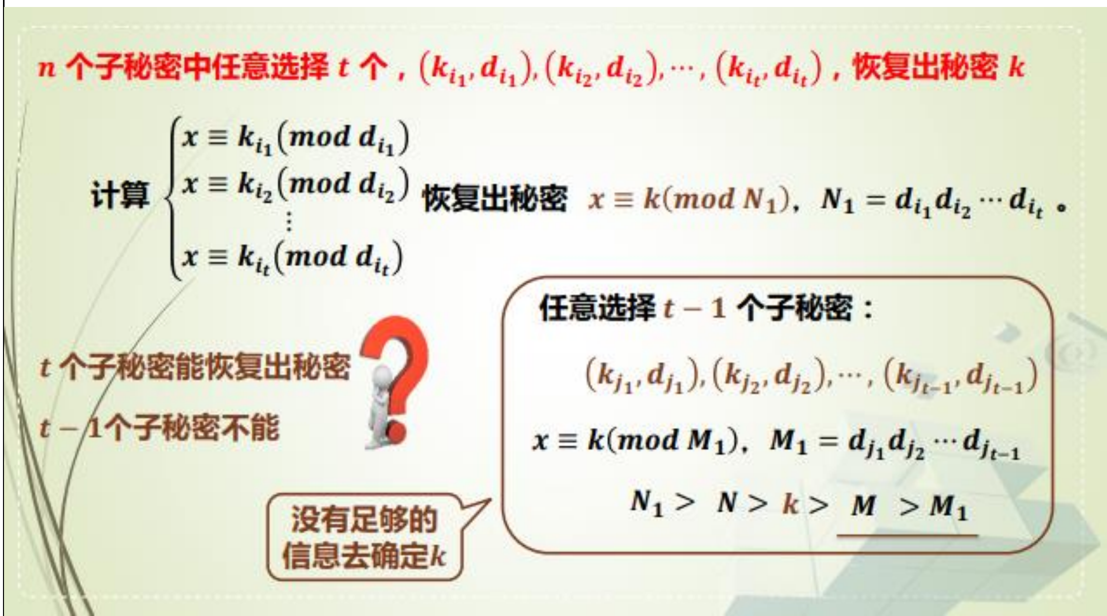


1983 年，Asmuth 和 Bloom 提出基于中国甚于定理的(t,n)门限秘密共享方案

原理：

高度概括原理就是使用同余方程进行加密，使用中国剩余定理来解密





三、方案实现

```c
#include"miracl.h"
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

#define T 3
#define S 5
#define RCV 3
#define OFFSET 7
#define BITS 500

int main(void)
{
    miracl  *mip = mirsys(1000, 16);

    int i = 0, j = 0, seed[RCV];//初始化
    big K, k[S], d[S], N, M, v;
    big D[S], D_1[S], n, x;

    K = mirvar(0); N = mirvar(1); M = mirvar(1);
    v = mirvar(1); n = mirvar(1); x = mirvar(0);
    for (i = 0; i < S; i++) {
        k[i] = mirvar(0); d[i] = mirvar(0);
        D_1[i] = mirvar(0); D[i] = mirvar(0);
    }

```

这部分代码的主要作用是导入 miracl 库，初始化变量。这里我们将秘密分成 5 份，t 设置成 3。

```c
    irand(time(NULL));
    puts("secret K=");//generate K
    bigbits(BITS, K);
    cotnum(K, stdout);

    expb2(BITS / T + OFFSET, v);//= 2^(BITS / T + OFFSET)
    incr(v, 1, v);    //= v+1              // find  d[i]
    for (i = 0; i < S; i++) {
        while (1) {
            incr(v, 2, v);//保证di严格递增
            if (isprime(v)) {
                printf("d[%d]= ", i);
                cotnum(v, stdout);
                copy(v, d[i]);
                break;
            }
        }
    }

```

这里我们使用 bigbits 函数随机生成需要加密的秘密，然后我们开始生成 d。这里我们采用了取巧的方法，我们先随机生成一个素数为 d1，然后在 d1 的基础上不停加 2，然后判断这个数是否为素数，如果是素数，就为 d2，依次往后。这样可以保证 d 之间是严格递增，且互素的。但是后来我们也发现了一些问

题，采用这种方法，会减少 d 的取值范围。

```c
	puts("N= ");
	for (i = 0; i < T; i++) {
		fft_mult(d[i], N, N);//N = d1 * d2 * d3 * d4 … … dt
	} cotnum(N, stdout);
	puts("M= ");
	for (i = 1; i < T; i++) {
		fft_mult(d[S - i], M, M);
	} cotnum(M, stdout);

	for (i = 0; i < S; i++) { //compute k[i]
		copy(K, k[i]);
		divide(k[i], d[i], d[i]);//ki = ki mod di
		printf("k[%d]= ", i);
		cotnum(k[i], stdout);
	}

	srand(time(NULL));//generate RCV random differente numbers
	for (i = 0; i < RCV; i++) {//choose num(RCV) random
		seed[i] = rand() % S;
		do {
			for (j = 0; j < i; j++) {
				if (seed[i] == seed[j]) break;
			}
			if (i != j) {
				seed[i] = rand() % S;
			}
			else {
				break;
			}
		} while (1);
	}

	// chinese remainder theory recover K
	for (i = 0; i < RCV; i++) {
		fft_mult(n, d[seed[i]], n);
	}
	printf("\nchoose %d k:\n", RCV);
```

加密过程，先计算 N，然后计算 k[i]，最后开始加密

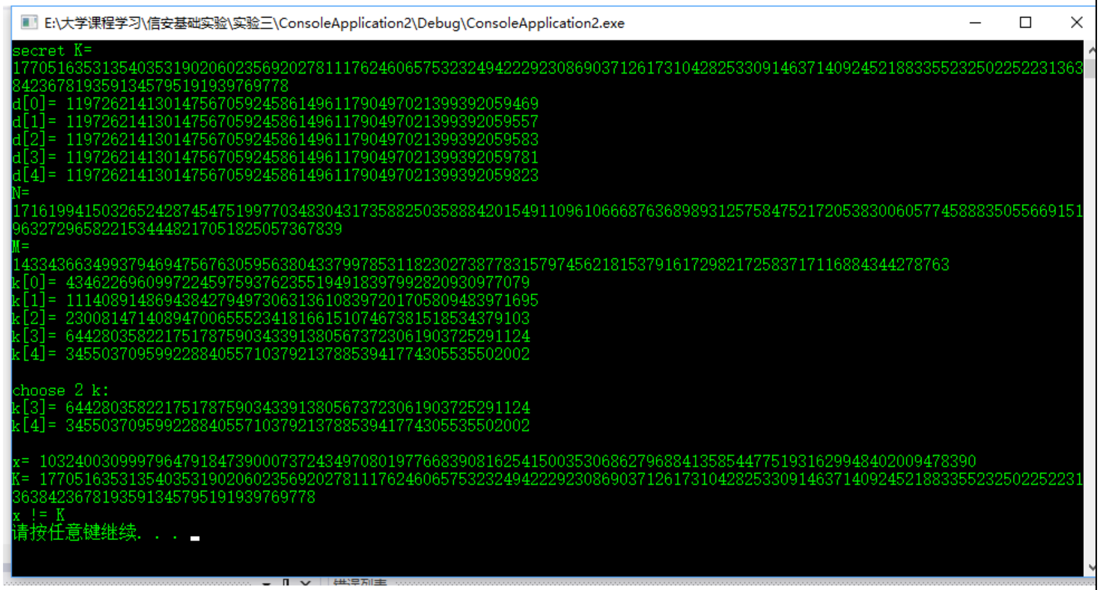
```c
	printf("\nchoose %d k:\n", RCV);
	for (i = 0; i < RCV; i++) {
		printf("k[%d]= ", seed[i]);
		cotnum(k[seed[i]], stdout);
		copy(n, v);
		divide(v, d[seed[i]], D[seed[i]]);
		invmodp(D[seed[i]], d[seed[i]], D_1[seed[i]]);
		fft_mult(k[seed[i]], D[seed[i]], d[seed[i]]);//k[seed[i]]*D[seed[i]]=d[seed[i]]
		fft_mult(d[seed[i]], D_1[seed[i]], d[seed[i]]);
		divide(d[seed[i]], n, n);
		add(d[seed[i]], x, x);
	}
	divide(x, n, n);

	putchar('\n');
	printf("x= "); cotnum(x, stdout);
	printf("K= "); cotnum(K, stdout);
	!mr_compare(x, K) ? puts("x == K") : puts("x != K");

	mirexit();
	system("pause");
	return 0;
}
```

解密过程

四、数据分析

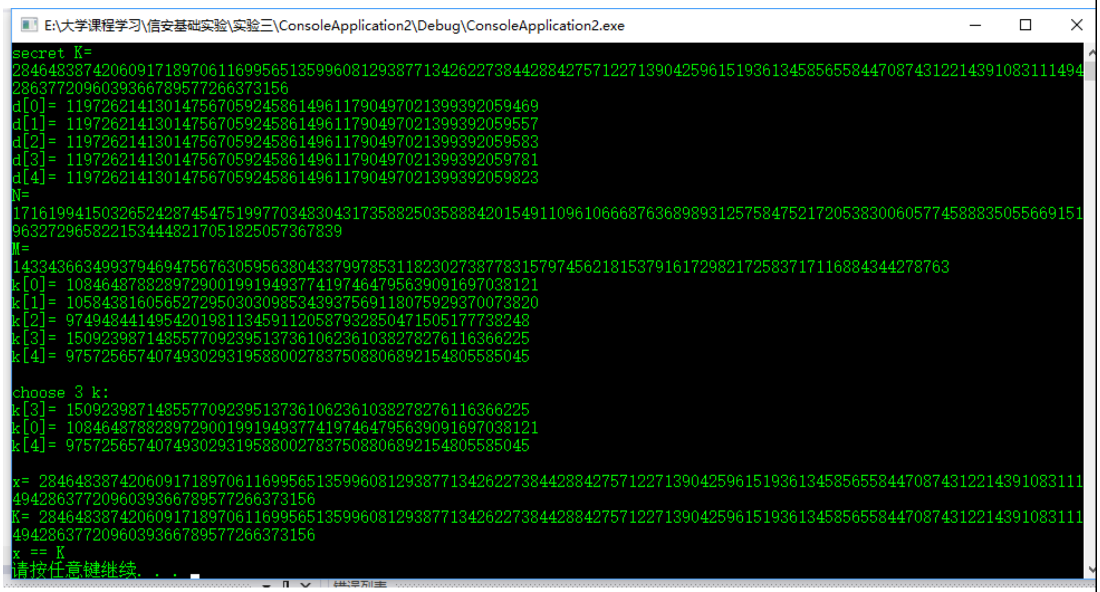
(包括算法测试数据的分析等等)

1.当 RCV 的值设置为 2 时（小于门限 3）:



发现不能成功解密出正确的消息。

2.当 RCV 的值设置为 3 时:



正确解密出了消息

五、总结

（完成的心得和其他，主要是自己碰到的问题，以及解决问题的方法等）

整个实验过程较为顺利，并没有出现较多的问题，主要是有上次中国剩余定理实验的积累，解密的过程并没有花费太多的时间。通过这个实验，我们也了解到了中国剩余定理在加密领域的应用