5/13/2013

Report A2 SSS

CSCI361

Tan Shi Terng Leon
4000602

Design

Share Generation

Getting the parameters

The user inputs t and m where t is the number of shares needed to reconstruct the secret and m is the total number of shares to generate

Generates a secret s

Generates a prime p where $p > \max(s, n)$

Constructing the equation

For each power of x in the polynomial equation, we generate a coefficient

$$y = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + \cdots + a_{t-1}x^{t-1} \ (mod\ p)$$

That is for $i = 0\ to\ t-1$ we generate a_i where $0 \leq a_i \leq t-1$ and a_0 is the secret

Stores the coefficients in an array

Generating the shares

For $x = 1\ to\ n$, we substitute the x value into the equation and get out corresponding y value

To do this, for each $i = 0\ to\ t-1$, we compute the $a_ix^i\ (mod\ p)$ and sum all the values together and (mod p) to get our corresponding y value (Note that $a_0x^0 = a_0\ mod\ p$ is actually the secret)

Hence we can compute n different shares of (x, y)

The information is saved in a file “Shares.txt”

Shares.txt

```
1  Secret is 228
2  p is 229
3  t is 3
4  n is 7
5  Equation is:
6  228 + 219x^1 + 158x^2
7  The shares are:
8  1,147
9  2,153
10 3,17
11 4,197
12 5,6
13 6,131
14 7,114
15
```

Share Reconstruction

Getting parameters

User inputs t, p and the shares. Some error checking is provided

Using Langrange interpolation to find the secret

There are t pairs of shares and we know that the polynomial equation is of degree t-1

For each share (x_k, y_k),

Compute numerator

We compute the numerator which is y_k multiplied by all $-x_j \; where \; j \neq k$

That is

$$numerator_k = y_k * \prod_{j=1, j\neq k}^{t} (-x_j)$$

Compute denominator

Now we compute the denominator that is,

$$denominator_k = \prod_{j=1, j\neq k}^{t} (x_k - x_j)$$

Get Secret

Then we calculate the inverse of the denominator and multiply it by the numerator, thus we have

$$value_k = numerator_k * \; {denominator_k}^{-1} \; (mod \; p)$$

For each pair of shares we have a value. Now we simply compute the same of all these values

$$\sum_{k=1}^{t} value_k$$

(Note: mod p still applies)

And thus we get out secret ☺

Program Manual

To run the program

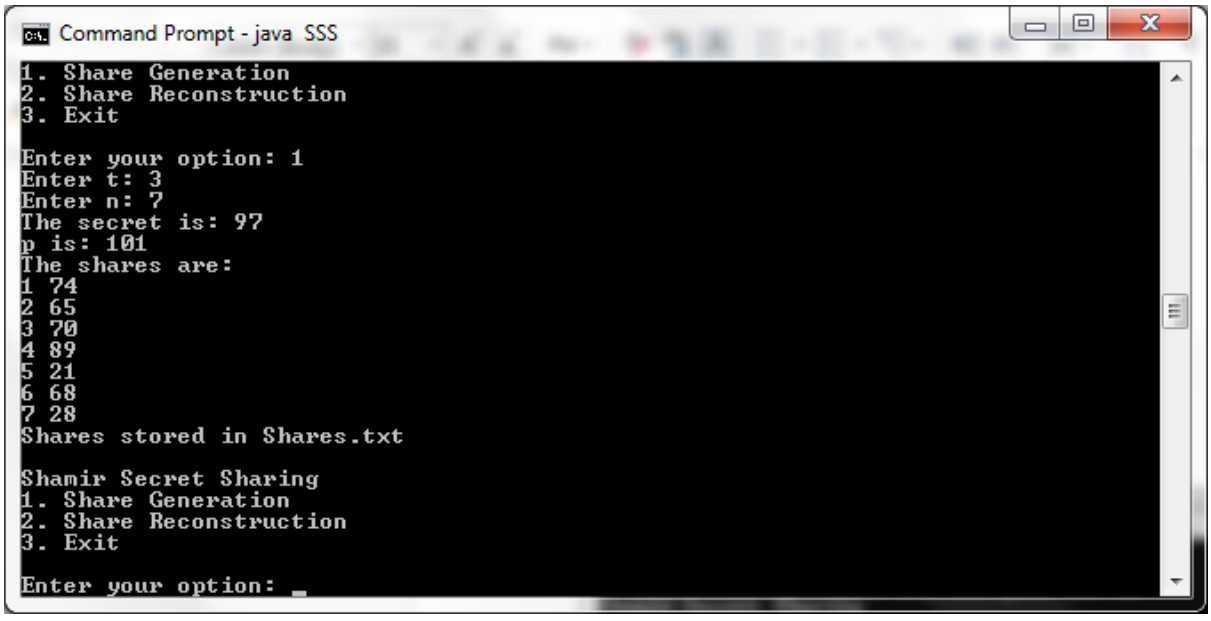
1. Cd to the directory containing the SSS.java
2. Enter “javac SSS.java”
3. Enter “java SSS”
4. Alternatively, to set the number of bits used to represent the integers used, enter “java SSS <size>” (the default is 8 bits)
    a. Eg, java SSS 16 (to use 16 bits integer)

Main Menu

```
C:\Users\User\workspace\SSS\src>java SSS
Shamir Secret Sharing
1. Share Generation
2. Share Reconstruction
3. Exit

Enter your option:
```

Share Generation

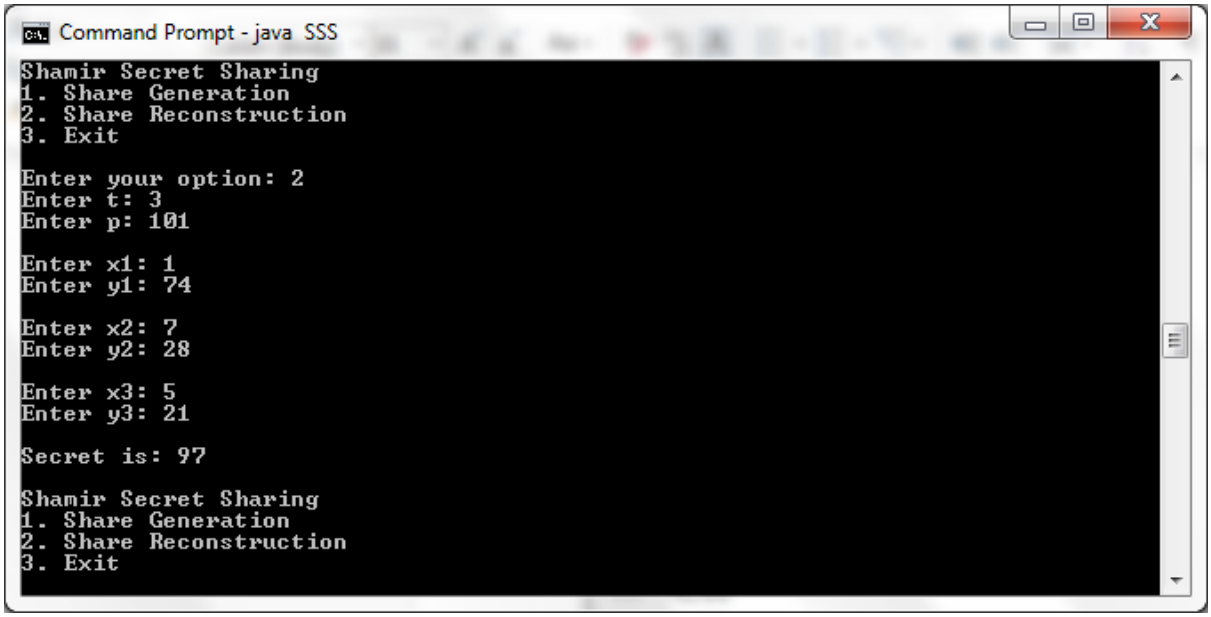


Shares.txt

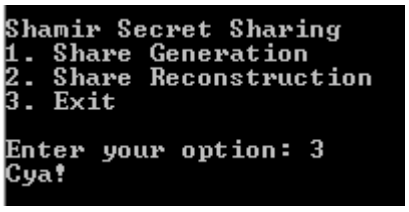
Shares.txt

```
1  Secret is 97
2  p is 101
3  t is 3
4  n is 7
5  Equation is:
6  97 + 71x^1 + 7x^2
7  The shares are:
8  1,74
9  2,65
10 3,70
11 4,89
12 5,21
13 6,68
14 7,28
15
```

Secret Reconstruction



Exit



Some error handling

```
C:\Users\User\workspace\SSS\src>java SSS
Shamir Secret Sharing
1. Share Generation
2. Share Reconstruction
3. Exit

Enter your option: 1
Enter t: 5
Enter n: 3
Please enter a value more or equal than the threshold (5)
Enter n:
```

```
Shamir Secret Sharing
1. Share Generation
2. Share Reconstruction
3. Exit

Enter your option: 1
Enter t: 0
Please enter a value more than 1
Enter t: -2
Please enter a value more than 1
Enter t:
```

```
C:\Users\User\workspace\SSS\src>java SSS
Shamir Secret Sharing
1. Share Generation
2. Share Reconstruction
3. Exit

Enter your option: 2
Enter t: 0
t must be greater than 1
Enter t: -12
t must be greater than 1
Enter t: 5
Enter p: 3
p must be greater than t (5)
Enter p: 124
p is not prime, please enter again
Enter p:
```