Advanced Security Lab 5 Report

by Tomas Belskis

C11477418

The first algorithm that was used to generate a random number is by the name of Wichmann-hill is a pseudorandom number generation algorithm. It takes in 3 seed numbers that are between 0 to 30000. It generates 3 random numbers using the the 3 seeds provided with the use of a set mathematical formula. When the 3 numbers are generated, they are put into another formula which return a random floating number between 0 to 1. In my program i used system time to generate 3 random seeds. I generate a BigInteger number by calling the wichmann-hill random number generator due to the number returned being a floating number i round it up to either 0 or 1 using Math.round(). I keep adding the 0s and 1s to my byte array until i reach 8bytes then i return a biginteger that has been created using my byteArray of random 0s and 1s.

The 2nd algorithm used is blum blum shub pseudorandom number algorithm takes a single seed, the seed that I generated using systemtime. And it takes 2 prime numbers, p and q, it computes the product of p and q into M. It takes the seed to the power of 2, I just multiply the seed by itself in the program and performs mod of M, then computed number is divided by M and this returns a floating number between 0 or 1. I use the same method for generating the big integer value as with the previous algorithm. By inserting the 0s and 1s into a byte array of size 8 once the byte array is filled with random number of 0s and 1s i create a biginteger based on the byte array.

In order to use the program, within the UI you click the button R - to generate a random number with the use of built in random number generation library. To generate a random number using Wichmann-hill algorithm you click the button AR1 then go to console and keep entering any integer value and pressing enter(this is to let the system time run and generate a random value, actual integer numbers aren't doing anything it's just for the purpose to slow down the system), after all the seeds have been generated you will get a random number(this might take a while due to each byte requiring 3 seeds and a total of 8 bytes is required to form a number). To use the blum blum shub alrgorithm you click on AR2 and go to the console and keep entering integer values for the same reason as stated above. After all the seeds have gone in you will get a random number. To generate a list of random numbers you click on the GenerateList button and go to the console to enter seeds as it uses the blum blum shub algorithm this might take a while due to having 10 numbers to generate. Once 10 numbers have been generated the list will be printed in the console. To check if the random number is prime you have to first generate a random number using one of the methods described above and click prime button, this shows where the number is prime or not within UI, to generate next prime you just click on the next prime button.

The difference between java library generated random number and numbers that had been generated by the algorithms, the only difference that i noticed is that it takes longer for the algorithms to generate a random number due to the fact that i have to generate seeds by having the user to input integer values but that's due to the way i have program my program otherwise i don't see any differences between java library generated and algorithm generated.