Stream Cipher
Task 2. Pseudorandom Number Generator

Murtadha Alobaidi

April 2022

Introduction

On this rapport, we Design our pseudo-random number generator based on the Linear Congruential Generator presented in the lecture "Symmetric key Encryption I".

Implementation

To design our pseudo-random, we need to choose our prime number "m". To find a value for "m" in a Linear Congruential Generator. There are different ways, that "m" is a prime or "m" is a power of 2.

We know if "m^2" can have a maximal period of m/4 and that can be limited depending on the size of the number. We chose "m a power of 2" to work with them on this task.

```
xi+1 = axi + b mod m
```

The benefit of "m a power of 2" is that allows the modulus operation to be computed by simply cat the binary representations.

```
int a = 48271;
int b = 12345678;
int m = 2147483647;
```

We choose the max prime number Integer that can have" $2^{31-1} = 2147483647$" and represented "m". When we need to represent a constant "a", we chose a value "48271". These values are used by a programming language "C++11's minstd_rand". The constant "b" can have any value because this does not change so much in the statistic.

Conclusion

When I run both programs and analyze my PRANG on “use Random.NextInt(256)” and “Random”. They look similar when it comes to the built Random on Java.