Implementation of a Pseudorandom Number Generator

CS290G - Cryptographic Engineering

John-Olav Storvold
johnolav.storvold@gmail.com

Johannes Omberg Lier
johannes.o.lier@gmail.com

May 30, 2015

Abstract

A good pseudorandom number generator (PRNG) is an essential ingredient in creating a secure cryptographic system. The output numbers of those number generators are used futher in encrypting and decrypting messages. Since the process of generating these numbers is deterministic a security system will have a major security breach if those numbers produced are predictable either by having a predictable output pattern or simply do not have a "random" enough seed input for it's number generation. The goal is to have an easy to produce pseudorandom number generator than generates numbers as close to random as possible. There is essentially a tradeoff between having a complicated good PRNG and having a PRNG that is fast, but also secure enough to use in production.

For our project we aim to create our own pseudorandom number generator which is fast in computation and generates reasonable results to be used in production. Our PRNG will have seed input gathered from various sources. The input seed will not be random, but will have an unpredictable pattern. Our motive is not to come up with a groundbreaking algorithm that exceeds already implemented algorithms, but to see if we can come up with an algorithm that is good enough to obtain some level of security.

When we have a working PRNG we will test the produced numbers and see if we have managed to create an algorithm that produces close to statistically random numbers.

Introduction