

Statistical Test for Randomness.

Venkata Koonaparaju

Specification initial submission

ECE 746

Introduction, Motivation and Originality

Most of the practical random number generators have some built in regularities which, sometimes, also become their deficiencies. Usage of these generators in cryptographic applications will need more security requirements than other applications. There are mathematical safety measures that offer some protection against this issue. One safety measure is to assess generators by theoretical tests and the other is to run empirical tests. Since the year 2000 NIST has been developing a string of statistical test to predict the efficiently predict the randomness of a string of bits. A total of 16 tests were developed, implemented and tested as of today. All of these tests were implemented in C by NIST. The purpose of this project is to implement at least 8 of the tests in C#. This project can be extended to implement all the 16 of them and can also be used in CrypTool 2.0 in future.

Language and Platform

This project will use .net framework 3.0 and will run on Microsoft windows platform. Visual Studio 2008 will be used to take advantage of new technologies like WPF and C# is used as the base language because of its popularity and modern concepts and design. Even though application developed in .net run mostly on windows platform there are applications out in the market like Mono 1.0 which will let these applications run on other platforms like Sun Solaris. This project doesn't look into the cross platform implementations and will use windows platforms.

Additional Software Required

This project uses the C code developed by NIST for reference purposes and also might use them for testing purposes.

Testing Strategy

The following steps will be involved in testing the implemented algorithms

- 1) Using a standard random number generator a set n bit sequences will be generated and stored in a text file
- 2) The tests implemented in C# will be run on the text file
- 3) A output file will be generated with relevant intermediate values and like P Values based on which conclusions can be drawn

The random strings generated for testing will consist of a mixture of periodic and almost random.

The empirical results can be conducted in many numbers of ways and this project uses the two approaches that NIST uses.

- 1) The proportion of sequences that pass a statistical test
- 2) The distribution of P-values to check uniformity.

Time Schedule

March – 24th, 26th

- Complete mathematical analysis of the 8 test to be implemented.
- Preparation of the development environment.
- Understand the C implementation of NIST

April -14th, 16th

- Finish the C# implementation of the almost all the test

April 28th, 30th

- Finish the C# implementation, if any, left out from the previous progress report
- Generate the test sequences and the out files
- Interpretation of implementation results.

Tentative Changes

Even though the obvious choice is to pick up the first 8 tests of the 16 the choice of tests might change and also the number of tests. The following is the list of tentative tests I am

1. The Frequency (Monobit) Test,
2. Frequency Test within a Block,
3. The Runs Test,
4. Test for the Longest-Run-of-Ones in a Block,
5. The Binary Matrix Rank Test,
6. The Discrete Fourier Transform (Spectral) Test,
7. The Non-overlapping Template Matching Test,
8. The Overlapping Template Matching Test

Tentative Table

1 Introduction

1.1 Overview of Randomness

1.1.1 Random Number Generators

1.1.2 Pseudorandom Number Generators

- 1.1.3 Testing
- 1.2 Definitions and Abbreviations.
- 1.3 Mathematical Symbols
- 2 Random Number Generation Test
- 3 Testing Strategy
 - 3.1 Different strategies of analysis
 - 3.2 Understanding the results
- 4 Conclusion

Literature

- 1) Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, San VoA, *Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, NIST Special Publication 800-22
- 2) www.cryptool.org
- 3) Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc., 1996