INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 07 ISSUE: 06 | JUNE - 2023 SJIF RATING: 8.176 ISSN: 2582-3930

Implementation of Test Suite for Randomness using MATLAB

1 Revansidda,2 Rohith kumar C J, 3 Sandeepa B Barangi,4 Vijay I Naik,5 Prof. L.Srividya,6 Dr. T.C.Manjunath

UG Students, BE(ECE), Electronics & Communication Engg. Dept., Dayananda Sagar College of Engineering, Bangalore
Assistant Professor, Electronics & Communication Engg. Dept., Dayananda Sagar College of Engineering, Bangalore
Professor & Head, Electronics & Communication Engg. Dept., Dayananda Sagar College of Engineering, Bangalore

Abstract

The project work undertaken by us involves the development of a test suite to check the randomness of data generated by random number source that is proposed to be used for cryptographic, modelling and simulation applications purposes. The statistical test suite by NIST, U.S. Department of commerce consists of 15 tests to select and test random number sources. It proposes criteria for characterizing and selecting appropriate generators. We are proposing to develop a test suite consisting of 14 feasible tests out of 15 tests from NIST test suite using MATLAB and implement it on a hardware kit. The random number source can be a standard generator like Bernoulli binary number generator, or output of an any cryptographic key generator algorithm. These generator types produce a stream of zeros and ones that may be divided into sub streams or blocks of random numbers. These are taken as input for our test suite, conduct tests on these data and randomness is evaluated. Our hardware will display the strength of the input binary sequence based on randomness and unpredictability and hence evaluates the suitability of random number source block for cryptographic applications. However, no combination of statistical tests can completely vouch for the suitability of a generator for use in a certain application; in other words, statistical testing cannot take the place of cryptanalysis. Generator design and cryptanalysis are outside the purview of our project..

Result

A 14 test implementation technique has been created, and MATLAB code for the 14 test has been written. It is confirmed and proven that the input data from a random number generator is unpredictable and random. The test suite's effectiveness was determined to be satisfactory. These evaluations could be helpful as a starting point for figuring out whether a generator is appropriate for a given cryptography, modelling, or simulation application.

Applications

Our study can provide a basic grasp of the NIST test programmes' process. Numerous cryptographic applications, such creating key material, can make use of the outputs of verified generators. The following applications of cryptography make use of random numbers: 1. Individual keys for use with digital signature algorithms 2. Parameters for protocols, such as those used in key establishment.

I. INTRODUCTION

The importance of random numbers in cryptographic applications is evident, as they are necessary for generating secure keys and providing random inputs for various cryptographic protocols.

To ensure the randomness of generated numbers, it is essential to perform randomness testing. This project aims to discuss the implementation of statistical tests for randomness, specifically focusing on random number sources used in cryptographic, modeling, and simulation applications.

The NIST has developed a set of 15 statistical tests that can effectively detect deviations from randomness in binary sequences generated by cryptographic random generators, whether they are hardware or software-based.

However, it is important to note that deviations from randomness can be attributed to either poorly designed generators or inherent anomalies in the tested binary sequences. Therefore, the tester must carefully analyze the test results to determine their significance and the appropriate interpretation.

For this project, we propose the development of a test suite using MATLAB, which will consist of 14 out of the 15 feasible tests from the NIST test suite. This suite will provide a comprehensive evaluation of the randomness of binary sequences and can be implemented on a hardware kit for practical applications.

II. LITERATURE REVIEWS / SURVEYS

[1] The paper titled "Byte-oriented Efficient Implementation of the NIST Statistical Test Suite" by A. Suciu, K. Marton, I. Nagy, and I. Pinca in 2020 proposes a highly efficient implementation of the NIST Statistical Test Suite by employing a byte-oriented approach and integrating various optimization techniques. This approach significantly reduces the execution time of the tests. By treating the individual bytes of the binary

sequence independently, the byte-oriented technique allows for parallel processing, leading to a substantial reduction in the overall execution time. Furthermore, this method also decreases the memory utilization of the implementation. The paper's optimization methods further enhance the effectiveness of the implementation.

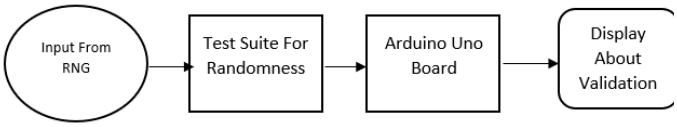
[2] The paper titled "A Comprehensive Review of Statistical Tests for Randomness and their Applications" by Qammar Abbas, Saima Javaid, and Muhammad Ishaq in 2020 provides a comprehensive analysis of various statistical tests for randomness, focusing specifically on their applications in the fields of communications and cryptography.

[3] The paper titled "A Comparative Study of Statistical Tests for Random number sources" by Sunil Kumar, Sonali Agarwal, and R. K. Agrawal in 2021 conducts a comprehensive comparison of multiple statistical tests for randomness, with a specific focus on evaluating their effectiveness and efficiency.

III. ENHANCEMENT

Our project leverages the power of MATLAB as the primary programming language for implementing the code. While other researchers may have utilized different programming languages for their respective projects, MATLAB proves to be particularly advantageous for our work due to its inherent capabilities in mathematical calculation programming. With MATLAB, we are able to efficiently handle complex mathematical operations, perform statistical tests, and analyze the randomness of binary sequences generated by various sources. By harnessing the computational capabilities of MATLAB, our project aims to enhance the accuracy and effectiveness of evaluating the randomness properties of random or pseudorandom number sources.

Block Diagram and Working:



Input data is taken from a random number generator.

These are taken as input for our test suite, conduct tests on these data and randomness is evaluated.

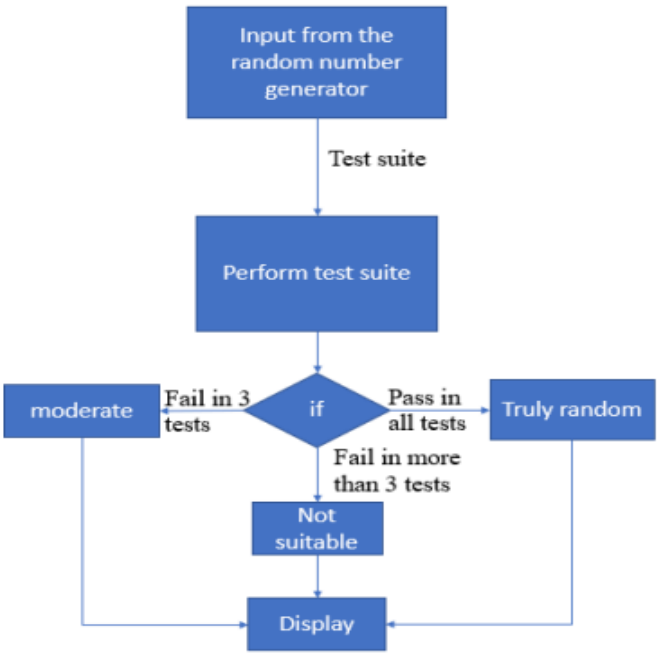
Our hardware will display the strength of the input binary sequence based on randomness and unpredictability and hence evaluates the suitability of random number generator block for cryptographic applications.

IV. PROPOSED METHODOLOGY

Implementing the design suite with MATLAB and Arduino Uno Board is the project's proposed work. The random/pseudo random number source can be the result of any cryptographic key generation process or a conventional generator like the Bernoulli binary number generator. These generator types generate a stream of zeros and ones that can be broken up into smaller streams or random number blocks. These are used as input for our test suite, which runs tests on them and assesses the randomness. Our technology will show the input binary sequence's strength based on randomness and unpredictability, allowing us to assess if the random/pseudorandom number source block is appropriate for use in cryptographic applications. However, no combination of statistical tests can completely vouch for the suitability of a generator for use in a certain application; in other words, statistical testing cannot take the place of cryptanalysis. Generator design and cryptanalysis are outside the purview of our project.

In summary, our project focuses on implementing the design suite using MATLAB and Arduino Uno Board to assess the randomness of random or pseudorandom number sources. By subjecting generated binary sequences to thorough statistical testing, we aim to evaluate their strength in terms of randomness and unpredictability. While statistical testing provides valuable insights, it is important to note that it cannot replace the comprehensive analysis performed through cryptanalysis.

Algorithm:



V. CONCLUSIONS

In conclusion, our project successfully evaluated the randomness and unpredictability of keys generated by a random number source. We improved the NIST statistical test suite for assessing the quality of random number sources and developed MATLAB code for the tests. Through simulations, we verified the unpredictability of the input data and obtained P-values for different binary sequences. Our project provides a reliable tool for determining the suitability of generators for cryptographic applications, enhancing the security and reliability of cryptographic systems.

REFERENCES

[1] Dong Lihua, Zeng Yong, Ji Ligang, Han Xucang, "Study on the Pass Rate of NIST SP800-22 Statistical Test Suite" in 10th International Conference on Computational Intelligence and Security, Kunming, China, 2014

[2] A. Suciu, K. Marton, I. Nagy, I. Pinca,"Byte-oriented Efficient Implementation of the NIST Statistical Test Suite", in IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR),Cluj-Napoca, Romania, 2014

[3] Haiqin Shi, Tao Pu*, Weifeng Mou and Yukai Chen,"NIST Randomness Tests on the Extended Key of Quantum Noise Random Stream Cipher", in 18th International Conference on Optical Communications and Networks, Huangshan China, 2019

[4] Emanuele Bellini, and Yun Ju Huang,"Randomness Testing of the NIST Light Weight Cipher Finalist Candidates",Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE,2021

[5] 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 sources for Cryptographic Applications",April 2010

[6] W. Stallings, "Cryptography and Network Security: Principles and Practice," 7th ed. Pearson, 2020.

[7] NIST Special Publication 800-22, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," National Institute of Standards and Technology, Gaithersburg, MD, USA, 2010.

[8] S. R. Vijayalakshmi and K. R. Ramakrishnan, "Enhanced Key Generation Scheme for Resource-Constrained Wireless Sensor Networks," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2986-2997, 2017.

[9] S. Kumar, R. S. Anand, and P. Kumar, "Efficient Key Management Scheme for Wireless Sensor Networks using Elliptic Curve Cryptography," IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 788-801, 2018.

[10] H. Kim, S. Park, and S. Park, "Lightweight Key Management Scheme for IoT Devices using Physical Unclonable Functions," IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5082-5093, 2020.

[11] S. Haykin, "Communication Systems," 5th ed. Wiley, 2013.

[12] https://store.arduino.cc/products/arduino-uno-rev3