

project proposal

by Osvaldas Stanelis

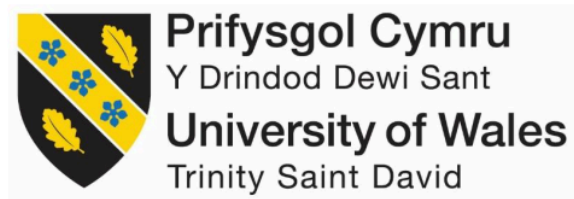
Submission date: 21-Oct-2024 01:32PM (UTC+0100)

Submission ID: 240762486

File name: 10895_Osvaldas_Stanelis_project_proposal_586419_584575766.docx (89.97K)

Word count: 2156

Character count: 13420



Design and Implementation of a Cryptographically Secure Pseudorandom Number Generator

by
Osvaldas Stanelis

Supervisor: NAME HERE

Project submitted as part of the requirements for the
award of

BSc (Hons) Computing (stream)
BEng (Hons) Electrical and Electronics Engineering
etc (change course name as applicable)

October 2024

Project title:**Design and implementation of a Cryptographically Secure Pseudorandom Number Generator (CSPRNG)****Identification of a specific problem:**

In the realm of cryptography, random number generation is foundational to secure communications and encryption processes. However, many current pseudorandom number generators (PRNGs) are not sufficiently secure for cryptographic purposes, making them vulnerable to attacks such as reverse engineering. Cryptographically Secure Pseudorandom Number Generators (CSPRNGs) provide randomness essential for encryption keys, secure tokens, and authentication mechanisms, ensuring the unpredictability needed to protect sensitive information. This project aims to address the issue of creating a reliable CSPRNG that balances high security with performance and scalability, suitable for small organizations that require robust data protection systems.

Well-defined aims:

The primary aim of this project is to design and implement a high-performance CSPRNG that can generate cryptographically secure random numbers. The CSPRNG will be evaluated for its randomness, security, performance, and scalability, ensuring it meets the requirements for practical cryptographic applications.

SMART objectives:

- **Specific:** Develop a secure, scalable CSPRNG that passes cryptographic randomness tests (e.g., NIST SP 800-22, Diehard) and meets high entropy standards.
- **Measurable:** The success of the CSPRNG will be measured by its performance in statistical randomness tests and its ability to resist common cryptographic attacks.
- **Achievable:** The project will leverage existing cryptographic libraries (e.g., OpenSSL) and implement secure algorithms, ensuring it can be completed within the project timeline.
- **Relevant:** As data security is crucial for modern organizations, the development of a secure PRNG is highly relevant to the field of cryptography and organizational cybersecurity.
- **Time-Bound:** The project will be completed within six months, with specific milestones for research, design, implementation, and testing.

Why the project is worth doing:

The increasing frequency of cyberattacks and data breaches highlights the need for strong encryption protocols, particularly in industries like finance, healthcare, and government. Many vulnerabilities arise due to weak or insufficiently random number generation, which can compromise encryption. By developing a secure and scalable CSPRNG, this project directly contributes to the protection of sensitive data, providing significant value to large organizations. It also offers the potential for broad application in the growing field of cybersecurity.

Project success criteria:

To succeed, the project must:

1. Ensure the CSPRNG passes recognized statistical randomness tests such as NIST SP 800-22 and Diehard.
2. Demonstrate high entropy, indicating resistance to predictability and cryptographic attacks.
3. Achieve acceptable performance benchmarks for speed and efficiency in generating random numbers.
4. Meet scalability requirements, proving adaptable for large-scale systems.
5. Deliver on time, with complete documentation and implementation ready for deployment.

Deliverables:

1. **CSPRNG Prototype:** A fully functioning, tested CSPRNG implemented in Python, optimized for performance.
2. **Statistical and Security Test Results:** A comprehensive report on the statistical randomness and security evaluation of the CSPRNG.
3. **Performance Report:** Documentation of the generator's efficiency, including generation speed and resource consumption metrics.
4. **Source Code and Documentation:** The complete codebase with clear guidelines for deployment, usage, and testing.
5. **Final Project Report and Presentation:** A summary of the project, including results, methodologies, and future recommendations for development.

Description of how the work relates to the literature review:

The work of this project builds upon the foundations laid in the literature regarding cryptographically secure pseudorandom number generators (CSPRNGs) and their role in encryption systems. Numerous studies have highlighted the vulnerabilities of traditional pseudorandom number generators (PRNGs) when used in cryptographic applications, particularly those that fail to meet entropy and unpredictability requirements. This project specifically addresses the security gaps identified in the literature, where weak PRNGs have led to successful attacks on cryptographic systems.

CSPRNGs have been widely researched in the context of secure communication, key generation, and cryptographic protocols. Several algorithms, including the Blum-Blum-Shub generator and Yarrow, have been proposed to increase randomness and unpredictability. However, many implementations trade off performance for security or fail to scale effectively for large systems. This project will seek to design a CSPRNG that not only meets cryptographic standards but also ensures high performance and scalability. The work will incorporate findings from previous studies that evaluated the statistical randomness and entropy of various algorithms, as well as recommendations for improving performance without compromising security.

Ethical issues:

There are minimal direct ethical concerns raised by this project, as it focuses on the design and implementation of a cryptographic tool. However, there are indirect ethical implications, particularly regarding the potential misuse of the technology. While the CSPRNG will be developed to protect sensitive data and ensure security, strong cryptographic tools can also be used by malicious actors to hide illegal activities. It is crucial that such tools are distributed responsibly, with careful consideration of the entities that may access and use them. In addition, responsible disclosure of any vulnerabilities discovered during the development process is essential to ensure the broader cryptographic community can respond to potential threats.

Identification of prior work and topic familiarity:

A review of key sources reveals that much of the existing work in CSPRNGs has focused on statistical randomness tests, entropy analysis, and algorithm efficiency. Studies such as those by Menezes et al. (1997) on the Handbook of Applied Cryptography, Kelsey et al. (1999) on entropy collection, and Ferguson and Schneier's (2003) work on Yarrow have provided foundational concepts in CSPRNG design and randomness testing. Other researchers like Koblitz and Menezes (2007) have focused on the mathematical underpinning of PRNG security, highlighting where failures can occur. This project will use their insights to avoid the vulnerabilities identified in previous work.

Literature review:

1. Menezes, A., van Oorschot, P., & Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.
2. Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (1999). Cryptanalytic attacks on pseudorandom number generators. *Proceedings of the 5th International Workshop on Fast Software Encryption*.
3. Ferguson, N., & Schneier, B. (2003). *Yarrow-160: Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator*. Springer.
4. Koblitz, N., & Menezes, A. (2007). Another look at "provable security". *Journal of Cryptology*.
5. Eastlake, D., & Crocker, S. (1994). Randomness Recommendations for Security. *RFC 1750*.
6. Matsumoto, M., & Nishimura, T. (1998). Mersenne Twister: A 623-dimensionally Equi distributed uniform pseudorandom number generator. *ACM Transactions on Modelling and Computer Simulation*.

Relevant models & concepts:

Key models used in this project include cryptographic security models that evaluate the unpredictability and entropy of the generated numbers. Additionally, randomness tests such as the NIST SP 800-22 and Diehard tests will be employed to ensure the generator meets recognized standards. Concepts of entropy collection and cryptographic security, as outlined in the works of Kelsey et al. (1999) and Schneier (2003), will be central to the project's evaluation criteria.

Project design

Development methods:

This project will use **Agile development methodology** to design and implement the cryptographically secure pseudorandom number generator (CSPRNG). Agile is an iterative, incremental approach that allows for flexibility in design, constant feedback, and regular adaptations to project requirements. This method is appropriate for several reasons:

1. **Incremental design:** Cryptographic algorithms require rigorous testing and adjustments based on continuous feedback. Agile allows the project to be broken down into manageable increments, where each iteration focuses on developing and testing key components of the CSPRNG (e.g., entropy sources, randomness tests, performance analysis).
2. **Flexibility:** The field of cryptography is evolving, and new vulnerabilities or best practices might emerge during the project timeline. Agile allows for adjustments in design and strategy without disrupting the entire project timeline, ensuring the CSPRNG can incorporate the latest cryptographic standards and solutions.
3. **Continuous testing and integration:** Cryptographic systems must be thoroughly tested at each stage of development. Agile encourages frequent testing and feedback, ensuring that any issues in the algorithm's randomness, performance, or security are identified early and addressed in the next development cycle.

Furthermore, **test-driven development (TDD)** will complement the Agile methodology. TDD emphasizes writing tests before writing the actual code, which is particularly useful in cryptographic systems to ensure that the algorithm meets its security and randomness objectives before further development proceeds.

Development tools:

To build the CSPRNG, the following tools and technologies will be used:

1. **Programming language:**
 - **Python** and **C** will be the primary programming languages used in this project. Python's extensive library support and readability make it a preferred choice for rapid prototyping and testing. Additionally, the well-established cryptographic libraries like *PyCryptodome* will help in the initial development and testing phases.
 - **C** will be used to implement the final CSPRNG version because it offers the low-level control necessary for optimizing performance and security, especially for cryptographic applications where memory management and speed are critical.
2. **Cryptographic libraries:**
 - **PyCryptodome** (Python) will be used for rapid prototyping and conducting early-stage entropy tests and randomness evaluations.
 - **OpenSSL** (C) will provide cryptographic functions and help in integrating the final CSPRNG with existing cryptographic libraries.
3. **Randomness testing tools:**

- **NIST SP 800-22** randomness test suite will be used to evaluate the randomness and statistical soundness of the generated numbers.
 - **Diehard tests** will also be applied to measure the quality of randomness.
 - These testing tools will ensure that the CSPRNG adheres to global standards for secure randomness.
4. **Version control:**
- **Git** will be used for version control to ensure the project's iterations are tracked effectively. This will also allow for collaborative work and easy rollback to previous versions in case a particular approach or modification is proven ineffective.
5. **Integrated development environment (IDE):**
- **Visual studio code** will be the main development environment for Python, while **CLion** will be used for C development. Both IDEs provide necessary debugging and plugin support for cryptographic development and are conducive to continuous integration workflows.

Data collection and analysis:

The data collected during this project will mainly consist of:

1. **Randomness test results:** These include statistical measures of the randomness of the generated numbers from NIST SP 800-22 and Diehard tests.
2. **Performance metrics:** Data on the CSPRNG's speed, memory usage, and scalability across various hardware environments will be gathered to ensure that the generator is efficient and suitable for large-scale deployment.
3. **Entropy source analysis:** The quality of the entropy sources will be evaluated based on their unpredictability and the absence of patterns or bias in the generated numbers.

Data analysis methodology:

- **Randomness analysis:** The test results from NIST and Diehard will be interpreted to ensure the CSPRNG produces statistically random output. This involves calculating p-values for various tests (e.g., frequency test, block frequency test, runs test) to ensure that the output follows a truly random distribution.
- **Performance analysis:** Collected data on execution time and resource consumption will be analysed using statistical tools to evaluate the generator's efficiency. The goal is to optimize for a balance between security and performance without introducing any vulnerabilities.
- **Security evaluation:** Using the collected test data, the CSPRNG will undergo cryptanalysis simulations to detect any patterns or vulnerabilities. This involves ensuring that the generator's output cannot be predicted even if part of the internal state is known.

Conclusion:

By combining Agile development with TDD and using tools like Python, C, NIST SP 800-22, and Diehard tests, this project ensures that the CSPRNG will be robust, efficient, and secure. The data collected during development will guide the optimization of the generator's randomness and performance, while continuous testing will ensure that it meets global cryptographic standards.

Resources/Planning

List of deliverables:

1. **CSPRNG prototype:** A fully functional cryptographically secure pseudorandom number generator implemented in Python (for prototyping) and C (for performance optimization).
2. **R Script for randomness testing:** R scripts developed for statistical analysis and visualization of the CSPRNG output, using packages like `ggplot2`, `stats`, and `random`.
3. **Statistical test results:** Reports on randomness and entropy, including NIST SP 800-22 and Diehard test outcomes.
4. **Security and performance report:** A comprehensive analysis of the CSPRNG's security against cryptographic attacks and its performance in terms of speed and resource usage.
5. **Source code and documentation:** Complete codebase with detailed documentation for the CSPRNG, including integration guidelines, user instructions, and R script usage.
6. **Final project report:** A project summary, including methodologies, results, and recommendations.
7. **Presentation:** A presentation showcasing the project's objectives, results, and final implementation.

Timing and resource use:

A Gantt chart will be created to illustrate the project timeline. The major phases include:

- **Research and design (November - January)**
- **Implementation and initial testing (January - March)**
- **Optimization and final testing (April)**
- **Documentation and reporting (May)**

Resources include Python, C, and R development environments (VS Code, CLion, RStudio), testing tools (NIST, Diehard), and Git for version control.

Ethics form:

The completed ethics form will address data confidentiality and responsible use of cryptographic technology, ensuring the project complies with ethical standards.

Appendices:

The appendices will include code snippets, detailed test cases, and relevant statistical analyses, including visualizations from the R script.

project proposal

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

89 / 100

PROJECT PROPOSAL – ACCA6020 INDEPENDENT PROJECT

Student **Osvaldas Stanelis**
Mark **89**

Intro / A&O (400 words)		
Project title; Identification of a specific problem to be investigated; Well-defined aims and SMART objectives; Explaining why this project is worth doing; Discussion of what the project needs to do in order to be a success; Description any deliverables that will be produced at the end of the project	20	20
The scope/aim of the project is identified and is well defined along with an outline of SMART objectives. The student outlines his interest in the topic to be researched and why he finds it worthwhile pursuing. The student also provides a template what steps need to be achieved in order for the project to be successful in his opinion, along with a list of		

measurable deliverables which can be produced at the project's completion.		
Background to the Project / Literature Review (400 words)		
Description of how the work to be done by in the project relates to the work identified in the literature review; Discussion of any ethical issues raised by the project Identifying what has already been done within the field; Topic familiarity; Brief literature review using appropriate sources – 6-8 journal papers and book chapters rather than just websites; Selection of relevant models & concepts A brief literature review of consisting of six articles comprising of papers from journals and academic literature has been undertaken, and does consider how previous work will impact on the current project. Ethical issues are commented upon and how their potential concerns regarding a misuse of the proposed algorithm for illegal activities and corruption. The student is familiar with previous work in this field but no	20	15

current/ recent publications are cited(2007 the latest). More in depth discussion of similarities and main findings of previous research in this field could have been outlined and related to the research project, with a view to how the student's project would be different to previous work, and possibly add to further knowledge in this field.		
Project Design (500 words)		
Clear statement of which development methods are to be used and why they are appropriate; Clear statement of which tools are to be used and why they are appropriate; Clear description of how data collected will be analysed The student provides a clear statement of the design of the project along with the development method and the relevant development tools, discussing their relevance and appropriateness in the completion of this investigation. The student also outlines the type of data to be collected as well as by what means, along with how the data will be analysed and the relevant statistical software tools and tests which need to be used in order to perform the analysis, and provide measures of randomness and entropy of the number sequences generated.	30	30

Resources / Planning (200 words)		
List of any deliverables; Clear diagram showing both timing and resource use; Completed ethics form; Limited and relevant appendices The Student provides a list of five deliverables of outcomes to be achieved at the end of the project. , No diagram or suitable chart indicating progress along a time line, ensuring the projects completion within the allocated time frame has been produced, but a brief outline of when each phase should be completed is provided. Although this has been included, it could have been more detailed and illustrative ensuring deadlines are met for timely completion of the project. The ethics form which is detailed in its content been submitted. Appendices in the form of robust computer code will be produced on completion of the project.	15	10

Document Presentation		
Correct spelling, grammar and punctuation; Flow of narrative/fluency; Coherence; Word processed ; Correct use of referencing with bibliography; Strict word limit The document is well written, is easy to read and flows appropriately. It is nicely word processed and presented. The referencing is appropriate and text supplied is within the word limit. Slight issues with coherence of the sections but on the whole well done.	15	14
TOTAL	100	89

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
