УДК 004.056.55

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КРИПТОГРАФИЧЕСКОЙ СТОЙКОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Хотеновская В.С.; Шахлан П.С. студенты

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Богдан Е.В. – маг. техн. наук, ассистент

Аннотация. В статье проводится сравнительный анализ криптостойкости генераторов псевдослучайных чисел: стандартного модуля random, защищенного модуля secrets и пользовательского LCG-алгоритма. На основе экспериментов показано, что высокие показатели энтропии и визуальная хаотичность данных не гарантируют защиту от предсказания последовательности. Работа наглядно демонстрирует уязвимость некриптографических генераторов к атакам восстановления состояния и обосновывает необходимость использования CSPRNG в задачах защиты информации, несмотря на снижение производительности.

Ключевые слова. Генератор псевдослучайных чисел, PRNG, CSPRNG, криптостойкость, информационная энтропия, алгоритм Mersenne Twister, линейный конгруэнтный генератор, информационная безопасность, Python, анализ алгоритмов, предсказуемость последовательностей.

Генераторы случайных и псевдослучайных чисел являются важным компонентом современных информационных систем, в особенности в области криптографии и информационной безопасности. Они применяются при формировании криптографических ключей, одноразовых токенов, nonce-значений, а также в различных протоколах аутентификации и шифрования.

Несмотря на широкое распространение, не все генераторы случайных чисел удовлетворяют требованиям, предъявляемым к средствам криптографической защиты. В частности, многие стандартные псевдослучайные генераторы (PRNG), используемые в языках программирования, ориентированы преимущественно на высокую производительность и приемлемые статистические характеристики, однако не обеспечивают достаточной криптографической стойкости, включая устойчивость к предсказанию и восстановлению внутреннего состояния [1, 2].

Использование подобных генераторов в задачах информационной безопасности может приводить к возникновению уязвимостей, связанных с возможностью восстановления последовательности генерируемых значений и прогнозирования последующих элементов. Это, в свою очередь, приводит к компрометации защищаемых систем.

В связи с этим актуальной является задача сравнительного анализа генераторов случайных и псевдослучайных чисел с учетом как их статистических характеристик, так и показателей криптографической стойкости.

Целью данной работы является сравнительное исследование генераторов псевдослучайных чисел, включая встроенные средства языков программирования и пользовательские алгоритмы, с точки зрения их пригодности для применения в криптографических задачах.

Для достижения поставленной цели решаются следующие задачи:

– рассмотрение теоретических основ функционирования генераторов псевдослучайных (PRNG) и криптографически стойких псевдослучайных чисел (CSPRNG);

– анализ характеристик выбранных генераторов;

– проведение статистического и криптографического тестирования;

– оценка устойчивости генераторов к практическим атакам.

Псевдослучайные генераторы (PRNG). Псевдослучайные генераторы чисел (Pseudo-Random Number Generators, PRNG) представляют собой алгоритмы, формирующие последовательности значений, обладающих статистическими свойствами случайных величин. Функционирование PRNG основано на детерминированном процессе, при котором каждое последующее значение определяется текущим внутренним состоянием генератора и начальным параметром – зерном (seed).

Одним из наиболее распространённых алгоритмов данного класса является Mersenne Twister, применяемый в стандартных библиотеках многих языков программирования [3]. Указанный алгоритм характеризуется высокой скоростью генерации и хорошими статистическими свойствами, включая равномерность распределения и большой период повторения. Внутреннее состояние генератора представлено массивом из 624 32-битных целых чисел. При наличии информации о состоянии становится возможным его полное восстановление, что позволяет предсказывать все последующие значения.

С точки зрения математической структуры Mersenne Twister относится к линейным рекуррентным генераторам. Линейность используемых преобразований обуславливает возможность восстановления внутреннего состояния на основе анализа выходной последовательности, в частности, с применением

методов решения систем линейных уравнений. Теоретические основы функционирования подобных генераторов подробно рассмотрены в работах Д. Кнута [4].

Ключевой особенностью PRNG является их детерминированность: при фиксированном начальном значении генератор воспроизводит одну и ту же последовательность. Данное свойство делает их удобными для задач моделирования и тестирования, однако ограничивает возможность применения в криптографических системах.

Кроме того, для многих PRNG разработаны методы восстановления внутреннего состояния на основе анализа части выходной последовательности. После восстановления состояния становится возможным предсказание последующих значений, что делает такие генераторы непригодными для использования в задачах, связанных с обеспечением информационной безопасности.

Криптографически стойкие генераторы (CSPRNG). Криптографически стойкие генераторы псевдослучайных чисел (Cryptographically Secure Pseudo-Random Number Generators, CSPRNG) предназначены для применения в условиях, предъявляющих повышенные требования к защите от предсказания и анализа генерируемых последовательностей.

В отличие от генераторов общего назначения, CSPRNG должны удовлетворять ряду строгих требований:

– непредсказуемость (unpredictability), заключающаяся в невозможности определения последующих значений на основе анализа части последовательности;

– устойчивость к восстановлению внутреннего состояния, при которой даже частичное раскрытие состояния не позволяет определить предыдущие или будущие значения;

– прямая криптографическая стойкость (forward secrecy), обеспечивающая невозможность восстановления ранее сгенерированных значений при компрометации текущего состояния;

– обратная криптографическая стойкость (backward secrecy), при которой знание предыдущих значений не позволяет предсказать последующие.

На практике CSPRNG реализуются на основе криптографических примитивов, включая блочные и потоковые шифры (например, ChaCha20), либо используют источники энтропии операционной системы, такие как /dev/urandom в Unix-подобных системах или специализированные криптографические интерфейсы в операционных системах семейства Windows.

В современных языках программирования для работы с криптографически стойкими генераторами, как правило, предусмотрены специализированные интерфейсы. В частности, модуль secrets языка Python использует системные источники энтропии и предназначен для генерации значений, применяемых в задачах обеспечения безопасности.

Таким образом, основное различие между PRNG и CSPRNG заключается в целевой направленности: первые ориентированы на обеспечение статистических характеристик и высокой производительности, тогда как вторые – на устойчивость к криптографическим атакам.

В качестве объектов исследования рассматриваются следующие генераторы:

1. Стандартный генератор общего назначения (random), основанный на алгоритме Mersenne Twister и широко применяемый в прикладных задачах, не связанных с криптографией.

2. Криптографически стойкий генератор (secrets), использующий источники энтропии операционной системы и предназначенный для генерации защищённых значений (токенов, ключей, паролей).

3. Пользовательский генератор, реализующий простой алгоритм псевдослучайной генерации (например, линейный конгруэнтный генератор или генератор семейства Xorshift), характеризующийся простотой реализации и высокой скоростью работы при ограниченной криптографической стойкости.

Для проведения сравнительного анализа генераторов псевдослучайных чисел разработана методология, включающая три взаимодополняющих направления: статистическое тестирование, криптографический анализ и оценку производительности.

Статистическое тестирование применяется для оценки соответствия выходной последовательности свойствам случайных величин. В рамках исследования используются следующие базовые тесты:

– частотный тест (monobit test), предназначенный для проверки равномерности распределения битов;

– тест серий (runs test), анализирующий длины последовательностей одинаковых битов;

– оценка энтропии, характеризующая степень неопределённости последовательности;

– критерий согласия χ^2, используемый для проверки соответствия эмпирического распределения теоретическому.

Указанные тесты позволяют выявить отклонения от случайности, однако их успешное прохождение не является достаточным условием криптографической стойкости генератора.

Криптографический анализ. Криптографический анализ направлен на оценку устойчивости генераторов к атакам, связанным с предсказанием выходной последовательности и восстановлением внутреннего состояния.

В рамках исследования рассматриваются следующие сценарии атак:

– атака восстановления состояния, при которой по части выходной последовательности предпринимается попытка реконструкции внутреннего состояния генератора;

– атака предсказания, направленная на определение возможности вычисления последующих значений на основе уже сгенерированных;

– анализ чувствительности к начальному значению (seed), предполагающий оценку влияния начальных параметров на предсказуемость последовательности.

Для стандартного генератора (Mersenne Twister) исследуется возможность восстановления внутреннего состояния по ограниченному числу выходных значений. Для пользовательского генератора (например, линейного конгруэнтного генератора) проводится анализ возможности восстановления параметров алгоритма.

Оценка производительности. Практическая применимость генераторов определяется, в том числе, их производительностью. В рамках исследования оцениваются следующие показатели:

– время генерации фиксированного объёма данных;

– среднее время получения одного значения;

– относительная нагрузка на вычислительные ресурсы.

Это позволяет сопоставить уровень безопасности генератора с его вычислительной стоимостью.

Условия проведения эксперимента. Экспериментальное исследование выполнено с использованием языка программирования Python. В качестве объектов анализа выбраны три реализации: стандартный модуль random (алгоритм Mersenne Twister), криптографически стойкий модуль secrets (интерфейс к системному CSPRNG), а также авторская реализация линейного конгруэнтного генератора с параметрами $a=1664525$, $c=1013904223$, $m=2^{32}$.

Для каждого генератора сформирована выборка объёмом $N=10^5$ значений. Оценка проводилась по трём критериям: статистические характеристики распределения, информационная энтропия и устойчивость к аналитическому восстановлению. Методология статистического анализа основана на подходах, изложенных в спецификации NIST SP 800-22 [5].

Результаты статистического анализа. Первичный анализ показал, что на уровне распределений исследуемые генераторы демонстрируют схожие характеристики. Гистограммы плотности распределения (см. рисунок 1) имеют близкий к равномерному вид без выраженных аномалий.

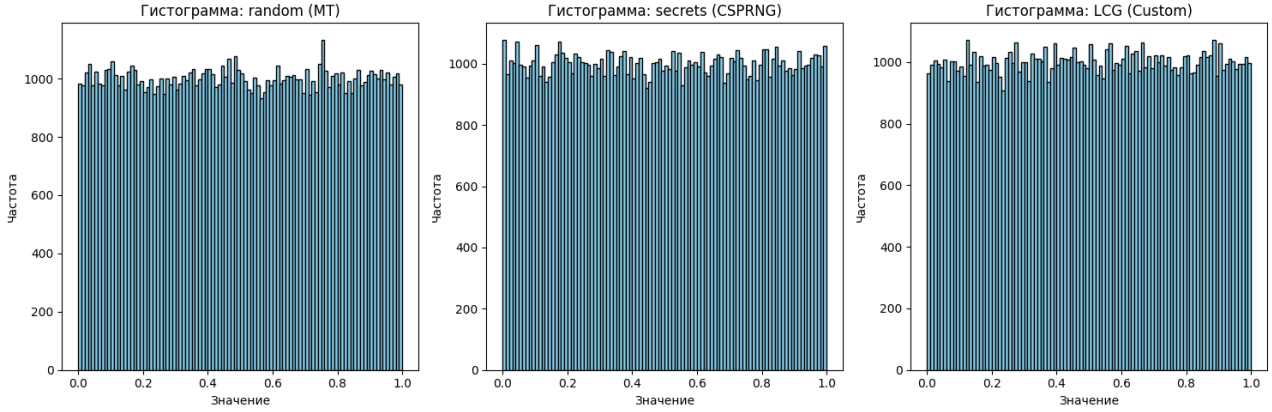

Рисунок 1 – Гистограммы распределения значений для исследуемых генераторов

Для количественной оценки вычислена информационная энтропия Шеннона по значениям, приведённым к 8-битному представлению. Для генератора random получено значение 7,9924, для secrets – 7,9926, для LCG – 7,9925. Близость указанных значений к теоретическому максимуму (8,0) свидетельствует о высокой степени неопределённости последовательностей.

Вместе с тем высокая энтропия отражает лишь качество статистического распределения и не является достаточным критерием криптографической стойкости, поскольку не исключает возможности предсказания последующих значений.

Результаты криптографического анализа. Проведённый криптографический анализ выявил принципиальные различия в стойкости исследуемых алгоритмов. Для визуализации скрытых зависимостей построены диаграммы рассеяния (scatter plot), в которых каждая точка соответствует паре соседних значений последовательности (см. рисунок 2).

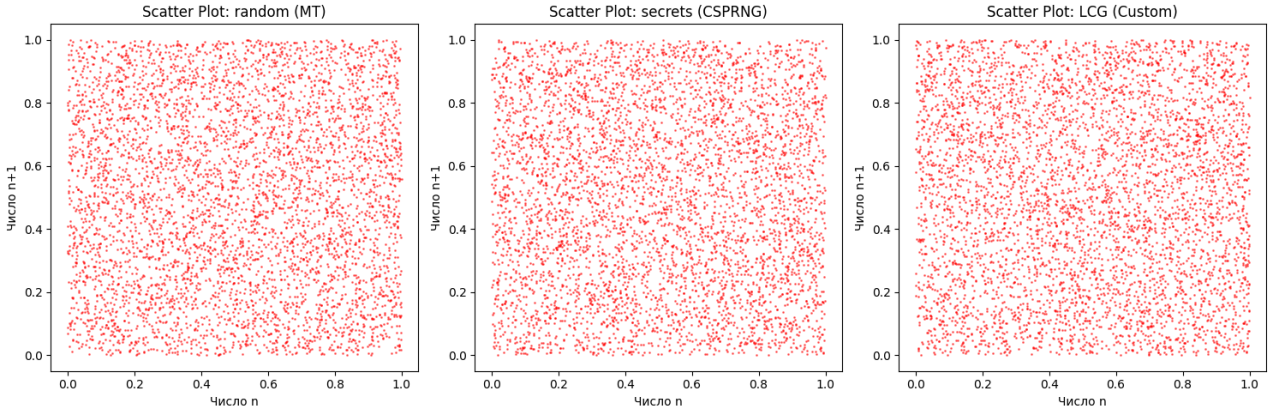

Рисунок 2 – Анализ корреляции соседних значений

Несмотря на визуально хаотичный характер распределения точек и отсутствие выраженных структур (см. рисунок 3), аналитические методы продемонстрировали уязвимость некриптографических генераторов.

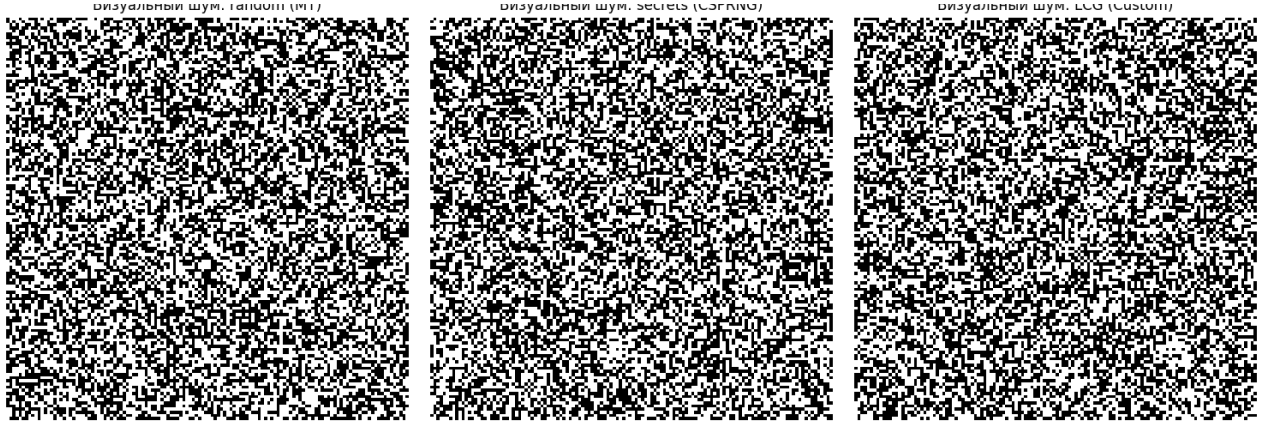

Рисунок 3 – Визуализация последовательностей в виде битового шума

В ходе эксперимента реализована атака на пользовательский генератор линейного конгруэнтного типа. Показано, что знание алгоритма и перехват одного текущего значения позволяют восстановить внутреннее состояние генератора. Для генератора Mersenne Twister установлено, что восстановление состояния требует 624 последовательных выходных значений, что также является практически реализуемым сценарием. В то же время попытки анализа генератора secrets не выявили статистически или аналитически значимых закономерностей.

Оценка производительности. Результаты измерения времени генерации 10^5 значений показали существенные различия в вычислительной эффективности алгоритмов. Установлено, что модуль secrets работает в 8,1 раза медленнее по сравнению со стандартным модулем random, тогда как пользовательская реализация LCG демонстрирует замедление примерно в 3,2 раза. Данные различия обусловлены использованием в CSPRNG дополнительных механизмов обеспечения безопасности, включая сбор энтропии и применение нелинейных криптографических преобразований.

Выводы по эксперименту. Проведённое исследование выявило существенное расхождение между статистическими характеристиками последовательностей и их криптографической стойкостью. Несмотря на близких к теоретически оптимальным значения энтропии и равномерности распределения, некриптографические генераторы продемонстрировали полную предсказуемость в условиях целенаправленного анализа. Показано, что стандартные методы визуальной и статистической оценки не позволяют выявить детерминированную природу PRNG, что ограничивает их применение в задачах информационной безопасности.

Установлено, что снижение производительности криптографически стойких генераторов является обоснованной платой за обеспечение устойчивости к аналитическим атакам, включая невозможность предсказания последующих значений.

Заключение. В работе выполнен комплексный сравнительный анализ генераторов псевдослучайных чисел различных классов, включая стандартные библиотечные реализации, криптографически стойкие генераторы и пользовательские алгоритмы. Экспериментально

подтверждена гипотеза о том, что статистическая случайность не является достаточным условием криптографической стойкости.

Показано, что генераторы общего назначения, такие как Mersenne Twister и линейные конгруэнтные алгоритмы, обеспечивают высокую скорость работы и приемлемые статистические свойства, однако обладают уязвимостями, связанными с возможностью восстановления внутреннего состояния. Это делает их непригодными для применения в защищённых информационных системах.

Практическая значимость работы заключается в обосновании необходимости использования специализированных криптографически стойких генераторов при решении задач, связанных с формированием ключей, токенов и иных чувствительных данных. Полученные результаты позволяют учитывать компромисс между уровнем безопасности и вычислительными затратами при проектировании информационных систем.

Список использованных источников:

1. Прикладная криптография. Протоколы, алгоритмы [Электронный ресурс]. – Электронные данные. Режим доступа: https://citforum.ru/book/cryptogr/fullsoder.shtml. Дата доступа: 05.03.2026.

2. Randomness Requirements for Security [Электронный ресурс]. – Электронные данные. Режим доступа: https://www.rfc-editor.org/rfc/rfc4086.html. Дата доступа: 05.03.2026.

3. Mersenne Twister [Электронный ресурс]. – Электронные данные. Режим доступа: https://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/ARTICLES/mt.pdf. Дата доступа: 05.03.2026.

4. Получисленные алгоритмы [Электронный ресурс]. – Электронные данные. Режим доступа: http://lib.ysu.am/disciplines_bk/843e2a65af2ae099da1a826e92f2d85b.pdf. Дата доступа: 05.03.2026.

5. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [Электронный ресурс]. – Электронные данные. Режим доступа: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf. Дата доступа: 05.03.2026.

UDC 004.056.55

A COMPARATIVE ANALYSIS OF THE CRYPTOGRAPHIC STRENGTH AND PERFORMANCE OF PSEUDO-RANDOM NUMBER GENERATORS

Khotenovskaya V.S., Shakhlan P.S., students

Belarusian State University of Informatics and Radioelectronics[1]
Minsk, Republic of Belarus

Bogdan E.V. – Master of Technical Sciences, Assistant

Annotation. This article presents a comparative analysis of the cryptographic security of pseudorandom number generators: the standard random module, the secure secrets module, and a custom LCG algorithm. Experimental results show that high entropy values and the apparent randomness of the data do not guarantee protection against sequence prediction. This work clearly demonstrates the vulnerability of non-cryptographic generators to state-recovery attacks and justifies the need to use CSPRNGs in information security applications, despite the resulting decrease in performance.

Keywords. Pseudorandom number generator, PRNG, CSPRNG, cryptographic strength, information entropy, Mersenne Twister algorithm, linear congruential generator, information security, Python, algorithm analysis, sequence predictability.