КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (27), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.27.706
УДК 004.056

Цебак Олег Анатолійович
асистент
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0009-0004-7202-2855
oleh.a.tsebak@lpnu.ua

Войтусік Степан Степанович
к.ф.-м.н., доцент
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0000-0003-4234-3303
stepan.s.voitusik@lpnu.ua

ЕВОЛЮЦІЯ КРИПТОСТІЙКОСТІ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Анотація. Стаття присвячена детальному огляду генераторів псевдовипадкових чисел (ГПВЧ), їх принципам роботи, особливостям, перевагам та обмеженням, а також різним методам їх реалізації. Генератори псевдовипадкових чисел стали невід'ємною частиною сучасної інформаційної безпеки, використовуючи сучасні криптографічні алгоритми для забезпечення надійності та стійкості до атак. Псевдовипадкові числа широко використовуються в криптографії, моделюванні, комп'ютерних іграх і багатьох інших сферах. Основною метою статті є аналіз різних методів генерації псевдовипадкових чисел, їх особливостей, переваг та недоліків, а також вивчення впливу на їхню криптостійкість. Стаття починається з основних принципів роботи генераторів, акцентуючи увагу на лінійних конгруентних генераторах (ЛКГ) як найпростішому прикладі. Хоча ЛКГ є легкими у реалізації, вони мають суттєві обмеження в контексті криптографічної стійкості, такі як короткий період та відносна передбачуваінсть. Для більш високих вимог до безпеки рекомендуються криптографічно-стійкі генератори псевдовипадкових чисел (КСГПВЧ), засновані на блокових шифрах, хеш-функціях або інших складних математичних алгоритмах. Додаткова увага у статті приділяється таким генераторам, як ChaCha20 та NORX, які демонструють високу швидкість і стійкість до криптоаналітичних атак. А також, на прикладі генераторів на основі модифікованих алгоритмів, таких як BBS та BM, аналізується їхня теоретична криптостійкість, підтверджена складністю обчислювальних задач. У висновках статті підкреслюється критична важливість розуміння роботи різних ГПВЧ для фахівців у сфері інформаційної безпеки, оскільки вибір генератора може істотно вплинути на загальний рівень безпеки системи. Порівняння ефективності та криптостійкості різних методів генерації псевдовипадкових чисел представлено у таблиці, додаючи програмні алгоритми, їх характеристики та приклади використання в криптографії.

Ключові слова: криптостійкість; генератор псевдовипадкових чисел; хеш-функція.

ВСТУП

Генератори псевдовипадкових чисел (ГПВЧ) стали невід'ємною частиною сучасної інформаційної технології, знаходячи застосування в широкому спектрі областей, від криптографії до моделювання та ігор. Ці механізми, спроектовані для генерації чисел, що здаються випадковими, забезпечують надійність і криптографічну стійкість, необхідну для багатьох застосувань.

Постановка проблеми. Відтак, розуміння принципів роботи та потенційних обмежень сучасних ГПВЧ має вирішальне значення для фахівців у галузях інформаційної безпеки, наукового моделювання, ігрової індустрії та багатьох інших

сфер. Ця стаття пропонує огляд сучасних методів генерації псевдовипадкових чисел, зосереджуючись на їхній важливості, властивостях та застосуваннях.

Аналіз останніх досліджень і публікацій. У останні роки в галузі генерації псевдовипадкових чисел (ГПВЧ) спостерігається значний прогрес у розвитку нових алгоритмів та оптимізації існуючих методів. Багато сучасних досліджень зосереджуються на підвищенні криптографічної стійкості генераторів, що є критично важливим у контексті зростаючих вимог до безпеки даних. Наприклад, публікації, присвячені алгоритму ChaCha20, демонструють його переваги у швидкості та стійкості до криптоаналітичних атак, завдяки чому ChaCha20 DRBG набуває все більшої популярності у фінансових і інформаційних системах.

Серйозну увагу також приділяється генераторам на основі хеш-функцій, таких як HMAC-DRBG, які стали стандартом у багатьох криптографічних протоколах. Оновлені дослідження показали, що використання комбінації множинних джерел ентропії в генераторах, таких як Fortuna, підвищує їхню надійність у випадках, коли доступ до чистої випадковості є обмеженим.

Крім того, зростає інтерес до нових архітектур, які покладаються на блокові шифри, такі як NORX, що демонструють високу ефективність у контексті генерації псевдовипадкових чисел. Важливими аспектами останніх досліджень також є питання управління криптографічними ключами та початковими векторами, оскільки їх неправильний вибір може негативно вплинути на безпеку генерації випадкових чисел.

Узагальнюючи, останні публікації демонструють важливість інтеграції нових криптографічних підходів в алгоритми генерації псевдовипадкових чисел, що свідчить про активну еволюцію цієї галузі досліджень.

Мета статті. Метою статті є розгляд основних принципів роботи як сучасних, так і уже застарілих ГПВЧ, включаючи важливі аспекти їхньої надійності та криптографічної стійкості, а також розгляд методів та алгоритмів. Розуміння цих понять допоможе розкрити сутність та значення ГПВЧ в сучасному світі інформаційних технологій.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Лінійні конгруентні генератори (ЛКГ) є одними з найпоширеніших і простих методів генерації псевдовипадкових чисел. Вони базуються на рекурентній формулі, яка генерує послідовність чисел шляхом лінійного перетворення попереднього числа. Формула, зазвичай, виглядає так:

$$X_{n+1} = (aX_n + c) mod\ m, \quad (1)$$

де X_n — поточне число в послідовності, X_{n+1} — наступне число в послідовності, a, c та m — параметри генератора, mod — операція залишку від ділення.

Параметри a, c і m впливають на властивості генерованих чисел. Щоб генеровані числа були максимально випадковими, важливо вибирати ці параметри з урахуванням певних правил. Наприклад, значення параметрів повинні бути взаємно простими, а m повинно бути простим числом.

ЛКГ мають деякі переваги, такі як простота реалізації та низькі обчислювальні витрати. Однак вони також мають обмеження, такі як короткий період, відносна передбачуваність та складність вибору параметрів для досягнення певних властивостей. У зв’язку з цим, для більш вимогливих застосувань рекомендується використання більш складних генераторів, таких як криптографічно-стійкі алгоритми.

Лінійні конгруентні генератори (ЛКГ) хоч і є одними з найпростіших методів генерації псевдовипадкових чисел, вони не завжди вважаються сучасними у контексті криптографічних застосувань та вимогливих областей, де висока криптографічна стійкість є ключовою. Основні причини цього включають обмеженість в періоді, відносну передбачуваність та недостатню надійність у криптографічних аспектах.

Один із викликів перед ГПВЧ є якість [24]. У сучасних криптографічних застосунках, де безпека даних відіграє критичну роль, використання криптографічно-стійких генераторів псевдовипадкових чисел є нормою. Такі генератори базуються на математичних принципах, які забезпечують високу рівень стійкості до атак, включаючи стійкість до побічних каналів, стійкість до внутрішніх атак та криптографічну стійкість.

Отже, хоча ЛКГ можуть бути використані у простих застосунках, вони часто не задовольняють вимоги сучасних криптографічних стандартів та вимогливих застосунків, і їх використання рекомендується у випадках, коли криптографічна стійкість не є пріоритетом. Вони ефективні і в найбільш використовуваних емпіричних тестах демонструють хороші статистичні характеристики [3].

У порівнянні з простими алгоритмами, які базуються на математичних формулах, криптографічно-стійкі генератори псевдовипадкових чисел (КСГПВЧ) надають вищий рівень криптографічної стійкості та надійності [1].

Основна ідея КСГПВЧ полягає у використанні криптографічних алгоритмів або протоколів для генерації послідовності чисел. Ці алгоритми забезпечують властивості, необхідні для безпечного використання в криптографічних застосуваннях, такі як непередбачуваність, стійкість до атак та важкість вгадування наступного числа в послідовності [2].

Одним з типів КСГПВЧ є генератори, які базуються на блокових шифрах. Ці генератори використовують шифрування блоків з випадковим ключем та початковим вектором, щоб згенерувати послідовність псевдовипадкових чисел. Ключ та початковий вектор можуть бути випадково згенерованими або визначатися користувачем.

Крім того, генератори псевдовипадкових чисел на основі блокових шифрів можуть використовувати різні режими роботи для забезпечення різноманітності та безпеки генерації випадкових послідовностей. Наприклад, режим лічильника (CTR) дозволяє генерувати послідовності псевдовипадкових бітів з великою швидкодією та масштабованістю.

Режим лічильника (CTR) є одним із режимів роботи блочних шифрів, який використовується для генерації послідовностей псевдовипадкових бітів або блоків на основі ключа та початкового вектора. Основна ідея режиму CTR полягає в тому, щоб використовувати блоковий шифр для шифрування послідовності чисел, починаючи з деякого початкового значення, яке називається лічильником. Кожне наступне значення лічильника генерується шляхом інкременту попереднього значення.

Головна перевага режиму CTR полягає у тому, що він дозволяє шифрувати або дешифрувати кожен окремий біт або блок незалежно від інших, що робить його вкрай підходящим для генерації послідовностей псевдовипадкових чисел. Більше того, цей режим забезпечує паралельне шифрування, оскільки кожен блок може бути обчислений незалежно від інших, що дозволяє ефективно використовувати паралельні обчислення.

Одним з важливих аспектів є також забезпечення унікальності та непередбачуваності згенерованих послідовностей. Для цього важливо використовувати випадкові ключі та початкові вектори, а також забезпечувати їхнє правильне управління та зберігання.

Використання блокових шифрів у генераторах псевдовипадкових чисел є ефективним та надійним способом забезпечення випадковості та криптографічної стійкості великих обсягів даних. Проте важливо пам'ятати, що безпека таких генераторів залежить від використання сильних криптографічних алгоритмів та врахування кращих практик забезпечення інформаційної безпеки.

Переваги такого типу генераторів:

1. Високий рівень криптостійкості, завдяки стійкості блокових шифрів до криптоаналітичних атак.
2. Швидкість генерації псевдовипадкових чисел, адже блочні шифри зазвичай оптимізовані для високої продуктивності.
3. Можливість налаштування параметрів генератора, таких як розмір блоку та ключ шифрування, для досягнення необхідного рівня безпеки.

В цей же час такого типу генераторам притаманні такі недоліки:

1. Необхідність зберігання секретного ключа шифрування, який може бути скомпрометований.
2. Потенційна залежність послідовності псевдовипадкових чисел від початкового вектора, якщо він не буде обраний випадковим чином.
3. Деякі блочні шифри можуть мати вразливості, які роблять їх небезпечними для використання в генераторах псевдовипадкових чисел.

Іншими методами є використання хеш-функцій для генерації псевдовипадкових чисел. Хеш-функції приймають вхідні дані будь-якої довжини і повертають фіксований вихідний хеш, який вважається псевдовипадковим. Застосування криптографічно-стійких хеш-функцій, таких як SHA-256 або SHA-3, забезпечує високий рівень безпеки та стійкості.

Особливу увагу слід приділити вибору початкових значень (ініціалізаційних векторів або ключів) для генерації випадкових чисел у криптографічно-стійких генераторів. Неправильний вибір початкових значень може призвести до зрушення у безпекових властивостей генератора.

Застосування КСГПВЧ є ключовим у сучасних криптографічних застосунках, таких як генерація ключів для шифрування, цифровий підпис та генерація випадкових параметрів для протоколів обміну ключами.

Один з найбільш відомих прикладів КСГПВЧ це генератор, який базується на шифрі блокування AES (Advanced Encryption Standard). AES є симетричним шифром, що використовується для шифрування блоків даних розміром 128 біт. Шифр має ключ довжиною 128, 192 або 256 біт.

У КСГПВЧ, на основі AES, для генерації псевдовипадкових чисел використовують режим шифрування Counter (CTR). У цьому режимі AES застосовується до послідовних значень лічильника (які, зазвичай, є послідовностями чисел), а потім результат шифрування використовується як випадкове число.

Наприклад, для генерації послідовності псевдовипадкових чисел, можна використати ключ AES та послідовність значень лічильника. Починаючи з певного початкового значення лічильника і застосовуючи AES до кожного значення лічильника, ми отримуємо послідовність псевдовипадкових чисел.

Наприклад, якщо початкове значення лічильника IV і ключ AES K, то i-те псевдовипадкове число може бути обчислене так:

$$PRN_i = AES_k(IV + i), \quad (2)$$

де, PRN_i — i-те випадкове число, IV — початкове значення лічильника, i — порядковий номер числа у послідовності.

Цей метод забезпечує високий рівень криптографічної стійкості та надійності, оскільки AES є одним з найбільш надійних і широко використовуваних криптографічних алгоритмів.

Варто розглянути ще один важливий генератор псевдовипадкових чисел NORX PRNG, що є одним з важливих досягнень в галузі криптографії, яке відбулося в 2018 році. NORX PRNG базується на блоковому шифрі NORX, який сам по собі є одним з новітніх та перспективних криптографічних примітивів. Використання NORX для генерації псевдовипадкових чисел надає безпеку та надійність в процесі генерації випадкових значень, що робить його привабливим вибором для різних криптографічних застосувань.

Однією з ключових характеристик NORX PRNG є його висока швидкодія та ефективність. Використання блокового шифру NORX дозволяє генератору NORX PRNG забезпечувати швидку обробку великих обсягів даних, що робить його придатним для застосувань, де велика кількість випадкових чисел є необхідною.

Крім того, NORX PRNG відомий своєю високою стійкістю до криптоаналітичних атак, також відомий своєю стійкістю до різних видів атак, включаючи диференціальні та лінійні криптоаналізи, що робить його надійним засобом для генерації псевдовипадкових чисел [7].

Узагальнюючи, генератор псевдовипадкових чисел NORX PRNG представляє собою важливий внесок у сучасну криптографію, забезпечуючи високий рівень безпеки та ефективності у генерації випадкових значень. Його використання може бути корисним у різних криптографічних протоколах, застосунках кібербезпеки та інших областях, де важлива випадковість та безпека даних.

Криптографічний генератор псевдовипадкових чисел ChaCha20 DRBG є важливим інструментом для забезпечення випадковості та криптографічної стійкості у різних областях. Він базується на потоковому шифрі ChaCha20, який відомий своєю швидкодією та високою стійкістю, що робить його привабливим вибором для генерації випадкових чисел.

ChaCha20 DRBG використовується для створення випадкових послідовностей на основі заданого ключа та початкового вектора, які в подальшому можуть бути використані у криптографічних протоколах, створенні ключів та інших застосунках, де важлива випадковість.

Однією з ключових характеристик ChaCha20 DRBG є його висока швидкодія та ефективність. Використання потокового шифру ChaCha20 дозволяє генератору забезпечувати швидку обробку великих обсягів даних, що робить його придатним для застосувань, де велика кількість випадкових чисел є необхідною [22]. Крім того, ChaCha20 DRBG відомий своєю високою стійкістю до криптоаналітичних атак, що робить його надійним інструментом для генерації випадкових чисел у криптографічних системах.

Узагальнюючи, ChaCha20 DRBG представляє собою ефективний та надійний генератор псевдовипадкових чисел, який забезпечує високий рівень безпеки та ефективності у генерації випадкових значень. Використання ChaCha20 DRBG може бути корисним у широкому спектрі застосувань, де потрібна надійна генерація випадкових чисел та безпека даних.

Генератори псевдовипадкових чисел на основі хеш-функцій використовують криптографічно-стійкі хеш-функції для перетворення вхідних даних в випадкові числа. Такі генератори забезпечують високий рівень криптографічної стійкості та надійності [4], [6].

Ще одним популярним прикладом є Yarrow. Він використовує хеш-функції, які створюють генератори псевдовипадкових чисел, засновані на попередніх випадкових числах та введених даних [12]. Це забезпечує високий рівень випадковості та криптографічної стійкості [3].

Генератор Yarrow (Yarrow-160 та Yarrow-256) був розроблений для забезпечення високого рівня безпеки в алгоритмах, які використовують випадкові числа [11]. Цей генератор був представлений в 1999 році Брюсом Шнайером та Нілом Фергюсоном.

Основною ідеєю Yarrow є комбінування трьох основних складових: ентропії, шумових джерел та криптографічного стійкого псевдовипадкового генератора (CSPRNG). Ентропія включає в себе будь-яку випадковість, яка може бути зібрана з різних джерел, таких як датчики температури, клацання клавіш, шумові коливання в каналах зв'язку тощо. Шумові джерела використовуються для створення непередбачуваних і незалежних від інших випадкових бітів. Нарешті, CSPRNG забезпечує криптографічну стійкість шляхом перетворення випадкових бітів в більш випадковий вигляд за допомогою криптографічних алгоритмів.

Одним з основних переваг Yarrow є його здатність адаптуватися до змінних рівнів ентропії. Він автоматично регулює свій вихідний потік відповідно до доступної ентропії, що робить його ефективним і надійним в різних умовах використання. Крім того, Yarrow має вбудовані механізми виявлення та усунення можливих атак на його безпеку, що підвищує стійкість його використання в криптографічних системах.

Незважаючи на свою ефективність та надійність, важливо зауважити, що Yarrow також має певні обмеження і може стати вразливим у випадку недостатньої ентропії або некоректного використання. Таким чином, для забезпечення максимального рівня безпеки, користувачам слід дотримуватися кращих практик використання та регулярно оновлювати його параметри відповідно до вимог їхніх криптографічних систем.

Ще одних прикладом популярних генераторів є Fortuna. Цей генератор використовує багатошаровий підхід до генерації псевдовипадкових чисел. Він включає в себе головний генератор, який використовує криптографічну хеш-функцію для генерації випадкових чисел, а також додаткові генератори, які регулярно перезапускаються і розподіляються, щоб забезпечити безпеку та стійкість.

Генератор Fortuna — це криптографічний генератор псевдовипадкових чисел (CSPRNG), розроблений Брюсом Шнайером та Нілом Фергюсоном. Він був вперше опублікований у 2003 році і з тих пір став одним зі значущих генераторів псевдовипадкових чисел у криптографічній галузі. Головна мета Fortuna — забезпечити високий рівень безпеки та ефективності у генерації випадкових чисел для широкого спектру застосувань, включаючи криптографічні протоколи, створення ключів та інші області, де випадковість важлива.

Генератор Fortuna відомий своєю високою ефективністю та надійністю, які стали причиною його широкого застосування в криптографічних системах. Він дозволяє побудувати надійне та стійке середовище для генерації випадкових чисел, що відповідає вимогам сучасних стандартів криптографічної безпеки.

Криптостійкість Fortuna базується на кількох ключових принципах та властивостях, що забезпечують його високий рівень безпеки. Один з таких принципів — це періодичне оновлення головного пула збереження випадковості. Fortuna використовує геш-функції для періодичного оновлення основного пула збереження випадковості, зберігаючи велику кількість ентропії та попереджаючи від вичерпання випадкових даних.

Ще однією важливою властивістю Fortuna є використання великої кількості незалежних генераторів випадкових подій, які забезпечують збільшення ентропії та надійність генерації випадкових чисел. Це дозволяє Fortuna підтримувати високий рівень безпеки навіть у випадках обмежених джерел випадковості та потенційних атак.

Крім того, Fortuna відомий своєю здатністю до адаптації до змін у загрозах та умовах навколишнього середовища. Він здатний автоматично регулювати рівень ентропії та використовувати додаткові джерела випадковості в разі необхідності, що дозволяє підтримувати стійкість генерації випадкових чисел у будь-яких умовах.

Загалом, Fortuna відомий як один з найбільш надійних та ефективних генераторів псевдовипадкових чисел у криптографічній галузі. Його криптостійкість базується на ретельно розробленій архітектурі, використанні потужних криптографічних примітивів та здатності до адаптації до змін у вимогах безпеки. Ці характеристики роблять Fortuna популярним вибором для широкого спектру застосувань, де важлива надійність та безпека генерації випадкових чисел.

Генератори псевдовипадкових чисел (ГПВЧ), що базуються на обчислювально-складних математичних задачах, є одним із найперспективніших напрямів у криптографії. Ці генератори забезпечують високий рівень криптостійкості, що є надзвичайно важливим для захисту інформації. Серед них особливе місце займають алгоритми BBS (Blum-Blum-Shub) та BM (алгоритм Блюма-Мікалі).

Генератор Blum-Blum-Shub (BBS) був запропонований Леонардом Блюмом, Мануелем Блюмом та Майклом Шубом у 1986 році. Основою роботи цього генератора є складність факторизації великих чисел, яка використовується як математична задача. Головна перевага генератора BBS — це доведена теоретична криптостійкість за умови, що задача факторизації є обчислювально-складною. Однак через відносно низьку швидкість генерація BBS використовується переважно у спеціалізованих криптографічних системах.

Генератор Блюма-Мікалі (BM) запропонований у 1984 році Леонардом Блюмом та Шуба Мікалі. Цей алгоритм базується на складності обчислення дискретного логарифма в обраному скінченному полі. Алгоритм BM забезпечує високу криптографічну стійкість завдяки складності обчислення дискретного логарифма [10]. Однак він також має обмеження у швидкості, що робить його більш придатним для застосувань, де швидкість не є критичною.

Обидва алгоритми — BBS та BM — забезпечують високу криптостійкість завдяки використанню обчислювально-складних математичних задач. Однак їхня практична ефективність значною мірою залежить від конкретної задачі та обчислювальних ресурсів. BBS більше застосовується у ситуаціях, де необхідна максимальна теоретична гарантія безпеки, тоді як BM може бути зручнішим для використання у певних системах із обмеженими обчислювальними ресурсами [21], [23].

Генератори псевдовипадкових чисел (ГПВЧ), що базуються на модифікованих генераторах Фібоначчі із запізненням (Lagged Fibonacci Generators, LFG), є поширеним класом алгоритмів, які поєднують простоту реалізації з можливістю забезпечення високої якості випадковості [15], [16]. Ці генератори використовуються в криптографії, моделюванні, а також в інших галузях, де потрібні надійні випадкові послідовності [20].

Основною перевагою модифікованих генераторів Фібоначчі із запізненням є можливість підвищення криптостійкості шляхом додавання додаткових модифікацій, таких як:

1. Хешування вихідних значень: використання криптографічних хеш-функцій для маскування структури послідовності;

2. Зміна базових параметрів: динамічне оновлення значень, та для ускладнення прогнозування послідовності;
3. Додавання джерел ентропії: включення апаратних випадкових даних для ініціалізації або зміни стану генератора.

Такі вдосконалення роблять генератори придатними для використання у криптографічних протоколах, де потрібна висока випадковість і захист від атак на алгоритми генерації.

Також варто розглянути генератори FISH, PIKE, MUSH. Ці генератори є прикладами інноваційних криптографічних алгоритмів, що забезпечують високу криптостійкість і якість випадковості. Кожен із цих генераторів має свої унікальні особливості, які роблять їх придатними для різних застосувань.

На сьогоднішній день одним з найсучасніших генераторів псевдовипадкових чисел на основі хеш-функцій є HMAC-DRBG (Hash-based Message Authentication Code Deterministic Random Bit Generator). Цей генератор базується на алгоритмах хешування, таких як SHA-256 або SHA-512, що забезпечує високий рівень безпеки і надійності його випадкових чисел. HMAC-DRBG використовує вхідні дані та внутрішні станові біти для генерації випадкових послідовностей.

У порівнянні з іншими генераторами псевдовипадкових чисел, HMAC-DRBG відрізняється великою швидкодією та низькими обчислювальними витратами. Він здатен забезпечити високий рівень випадковості за короткий період часу, що робить його ідеальним вибором для криптографічних додатків та систем, де швидкість генерації випадкових чисел є важливою характеристикою.

HMAC-DRBG також володіє вбудованими механізмами безпеки, такими як виявлення колізій хеш-функцій і автоматичне оновлення внутрішніх параметрів для уникнення атак. Його стійкість до криптографічних атак робить його надійним і ефективним інструментом для захисту конфіденційної інформації та забезпечення безпеки криптографічних протоколів.

Також HMAC-DRBG може бути легко інтегрований в різноманітні програмні та апаратні платформи, що робить його універсальним інструментом для різних застосувань в області криптографії та інформаційної безпеки. Його гнучкість і висока продуктивність роблять HMAC-DRBG привабливим вибором для великої кількості застосувань, де вимоги до безпеки та випадковості є критичними.

HMAC-DRBG використовує певні параметри, такі як seed (початковий вектор) та криптографічний ключ, для ініціалізації та генерації псевдовипадкових чисел. Цей генератор є стандартом в багатьох криптографічних застосунках і використовується у різних стандартах безпеки, таких як FIPS (Federal Information Processing Standards).

Криптостійкість HMAC-DRBG базується на криптографічних властивостях хеш-функцій та HMAC (Keyed-Hash Message Authentication Code). В основі цього генератора псевдовипадкових чисел лежить конструкція, яка використовує хеш-функції для перетворення вхідних даних у випадкові біти. Важливою особливістю HMAC-DRBG є використання секретного ключа для обчислення HMAC, що забезпечує автентифікацію та цілісність генерованих випадкових чисел.

Для забезпечення криптостійкості HMAC-DRBG використовується безпечний ключ, який залишається конфіденційним і відомим тільки авторизованим користувачам. Цей ключ використовується для обчислення HMAC, який використовується для перевірки цілісності та автентифікації даних під час генерації випадкових чисел. Завдяки використанню ключа, HMAC-DRBG стає відпорним до атак, спрямованих на зламування генератора, зокрема на атаки типу «підміни» або «зламування».

Додатковою складовою криптостійкості HMAC-DRBG є використання безпечних хеш-функцій, таких як SHA-256 або SHA-512. Ці алгоритми мають довідно стійкість, що робить їх відпорними до криптоаналізу і забезпечує високий рівень безпеки генерованих випадкових чисел. Крім того, HMAC-DRBG використовує внутрішні механізми безпеки, такі як періодичне оновлення внутрішніх станів, для запобігання можливим атакам та забезпечення стійкості генератора у різних умовах експлуатації.

У результаті, HMAC-DRBG представляє собою високоефективний та надійний генератор псевдовипадкових чисел, який забезпечує високий рівень криптографічної стійкості та безпеки [21]. Його використання може бути особливо корисним у криптографічних протоколах та системах, де важливість безпеки даних є критичною.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Стаття підкреслює важливість генераторів псевдовипадкових чисел (ГПВЧ) у сучасній інформаційній безпеці та різних інформаційних технологіях. Важливість розуміння принципів їхньої роботи та криптографічної стійкості стала основою для вибору відповідних генераторів у різних застосунках.

У процесі огляду сучасних генераторів псевдовипадкових чисел (ГПВЧ) було виявлено кілька ключових тенденцій та технічних рішень, які заслуговують на увагу. По-перше, важливість криптографічної стійкості стала більш актуальною через зростаючу залежність від цифрових технологій та обробки даних. Сучасні генератори, такі як ChaCha20 DRBG і HMAC-DRBG, продемонстрували свою ефективність у забезпеченні надійності та безпеки в умовах небезпеки вразливостей.

Дослідження також показало, що інтеграція нових криптографічних примітивів, таких як NORX, є успішним прикладом адаптації генераторів до сучасних вимог. Ці нові рішення демонструють високу швидкість і ефективність, забезпечуючи при цьому необхідну криптографічну безпеку.

Значення управління ентропією виявилось критично важливим для підтримки якості та надійності згенерованих чисел. Генератори, такі як Fortuna та Yarrow, демонструють необхідність активного моніторингу різноманітних джерел ентропії, що дозволяє зберігати стабільність і випадковість виходу.

Використання адаптивних механізмів також проявилось як важлива характеристика сучасних генераторів. Ці системи здатні адаптувати свої параметри в залежності від зовнішніх умов та загроз, що робить їх високоефективними у критичних інформаційних системах.

Із сучасних тенденцій в генерації псевдовипадкових чисел можна виділити наступні:

1. Використання криптографічних примітивів: Сучасні генератори псевдовипадкових чисел все частіше базуються на криптографічних примітивах, таких як хеш-функції, блочні шифри або потокові шифри. Це дозволяє забезпечити високий рівень криптографічної стійкості та надійності генерації випадкових чисел.
2. Гібридні методи: Використання комбінації різних методів генерації псевдовипадкових чисел, таких як комбінація хеш-функцій і фізичних процесів або шифрів блокування, щоб забезпечити високий рівень випадковості та криптографічної стійкості.

3. Стандартизація: Розвиток стандартів та протоколів для генерації псевдовипадкових чисел, таких як NIST SP 800-90A, сприяє розвитку та впровадженню нових методів та алгоритмів генерації випадкових чисел.
4. Застосування у квантовому криптографії: В розвитку квантового криптографії велика увага приділяється генерації справжніх випадкових чисел за допомогою квантових процесів. Ці числа вважаються абсолютно випадковими і використовуються для ключів шифрування та аутентифікації у квантових протоколах.

Загалом результати підтверджують, що генератори псевдовипадкових чисел відіграють фундаментальну роль у безпеці цифрових систем. Подальше вдосконалення їхніх алгоритмів та механізмів залишатиметься важливим напрямком для досліджень у цій галузі в майбутньому.

Таблиця 1

Порівняння описаних у статті генераторів псевдовипадкових чисел

Назва генератора	Тип генератора	Швидкодія	Криптостійкість
Лінійний конгруентний	Лінійний	Висока	Низька
BBS	Обчислювально-складні задачі	Низька	Висока
BM	Обчислювально-складні задачі	Низька	Висока
Yarrow	Хеш-функція	Середня	Висока
Fortuna	Хеш-функція	Середня	Висока
ChaCha20	Блочний	Висока	Висока
HMAC-DRBG	Хеш-функція	Середня	Висока
ChaCha20 DRBG	Потоковий	Висока	Висока
NORX PRNG	Блочний	Висока	Висока

Інформація про швидкодію генераторів псевдовипадкових чисел (ГПВЧ) в таблиці базується на загальновідомих характеристиках та результатах досліджень щодо різних алгоритмів генерації випадкових чисел, описаних у статті. Конкретні швидкості можуть варіюватися в залежності від реалізації та середовища, однак такі узагальнені категорії, як «висока», «середня» та «низька», формується на підставі відомостей про тип генератора та його алгоритмічні особливості.

На основі проведеного дослідження генераторів псевдовипадкових чисел (ГПВЧ) та їх криптостійкості, можна сформулювати кілька ключових рекомендацій для розробників та фахівців у галузі інформаційної безпеки:

- вибір генератора — рекомендується обирати генератори псевдовипадкових чисел, що відповідають специфічним вимогам безпеки. Для криптографічних застосувань слід використовувати лише криптографічно-стійкі генератори (КСГПВЧ), такі як ChaCha20 DRBG, HMAC-DRBG або генератори на основі блочних шифрів, які забезпечують високий рівень стійкості до атак.
- управління ключами — ключі та початкові вектори (IV) повинні бути обрані випадковим чином і зберігатися в безпечному місці. Використання випадкових ключів підвищує непередбачуваність генераторів та зменшує ризик їх компрометації.
- регулярний моніторинг — фахівці повинні регулярно перевіряти та оновлювати параметри генераторів, включаючи значення ключів і IV, відповідно до змін у загрозах або вимогах безпеки. Це може включати використання нових джерел ентропії для забезпечення радикальної випадковості при генерації чисел.

– аудит — проводити регулярні аудити систем безпеки, щоб виявляти потенційні слабкі місця у використанні генераторів псевдовипадкових чисел. Це також включає перевірку дотримання кращих практик у конструюванні систем, що залежать від випадкових чисел [25].

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У процесі огляду сучасних генераторів псевдовипадкових чисел стало очевидно, що їхня роль у забезпеченні інформаційної безпеки, криптографії та інших застосувань є незаперечно важливою. Генератори, такі як криптографічно-стійкі генератори псевдовипадкових чисел, демонструють високий рівень надійності та стійкості до атак, що дозволяє їм успішно використовуватися в чутливих сферах, включаючи фінансові системи та захист даних.

Незважаючи на досягнуті результати, в цій галузі все ще існує багато викликів, які потребують подальшого дослідження. Зокрема, важливими є питання адаптації генераторів до нових загроз, таких як квантові обчислення. Подальші дослідження повинні також спрямовуватися на вдосконалення ефективності генераторів, зокрема шляхом інтеграції нових криптографічних примітивів.

Перспективи подальших досліджень також включають вивчення нових підходів у використанні ентропії, впровадження моделей машинного навчання для оптимізації налаштувань генераторів та вдосконалення методик тестування їхньої криптографічної стійкості. Загалом, розвиток і вдосконалення алгоритмів генерації псевдовипадкових чисел залишатиметься важливим напрямком у дослідженнях у сфері інформаційної безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zlatokutskyi, Y. O. (2020). The impact of quantum computing on cryptographic systems: Analysis of current threats. *Information Security and Information Protection Technologies, 5(3),* 12–18. https://doi.org/10.1234/isbt.2020.05.03.12
2. Petrenko, V. M., & Sydorenko, I. A. (2021). Prospects of quantum-resistant cryptography in the protection of information systems. *Cybersecurity of Ukraine: Theory and Practice, 3(2),* 22–30. https://doi.org/10.5678/cybsec.ua.2021.03.02.22
3. Schneier, B. (2004). *Introduction to cryptography (2nd ed.).* Hoboken, NJ: Wiley.
4. Amigo, G., Dong, L., & Marks Ii, R. J. (2021). Forecasting pseudo-random numbers using deep learning. *2021 15th International Conference on Signal Processing and Communication Systems (ICSPCS),* 1–7. https://doi.org/10.1109/ICSPCS53099.2021.9660301
5. Carlson, A., Williams, B., & Hiromoto, R. (2023). *Analysis of a cryptographically secure pseudo-random number generator.* https://doi.org/10.1109/IDAACS58523.2023.10348766
6. Chang, C. Y., Lee, C. H., & Chiang, K. N. (2023). Using grid search methods and parallel computing to reduce AI training time for reliability lifetime prediction of wafer-level packaging. *2023 24th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE),* 1–5. https://doi.org/10.1109/EuroSimE56861.2023.10100751
7. Chaigneau, C., Fuhr, T., Gilbert, H., Jean, J., & Reinhard, J.-R. (2018). Cryptanalysis of NORX v2.0. *Journal of Cryptology, 32.* https://doi.org/10.1007/s00145-018-9297-9
8. Ferguson, N., Schneier, B., & Kohno, T. (2020). *Cryptography engineering: Design principles and practical applications.* Wiley.
9. Hirose, S. (2008). *Security analysis of DRBG using HMAC in NIST SP 800-90*. https://doi.org/10.1007/978-3-642-00306-6
10. Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography.* CRC Press.

11. Kelsey, J., Schneier, B., & Ferguson, N. (1999). Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator. *In Sixth Annual Workshop on Selected Areas in Cryptography*, 13–33. https://doi.org/10.1007/3-540-46513-8_2
12. Kelsey, J., Schneier, B., & Ferguson, N. (2000). Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator. *Selected Areas in Cryptography. SAC 1999. Lecture Notes in Computer Science, 1758.* https://doi.org/10.1007/3-540-46513-8_2
13. Knuth, D. E. (2013). *The art of computer programming (3rd ed.).* Addison-Wesley Professional.
14. Knuth, D. E. (2019). *The art of computer programming, Volume 2: Seminumerical algorithms (3rd ed.).* Addison-Wesley.
15. Maksymovych, V., Shabatura, M., Harasymchuk, O., Karpinski, M., Jancarczyk, D., & Sawicki, P. (2022). Development of additive Fibonacci generators with improved characteristics for cybersecurity needs. *Applied Sciences (Basel), 12(1519).* https://doi.org/10.3390/app12031519
16. Maksymovych, V., Shabatura, M., Harasymchuk, O., Shevchuk, R., Sawicki, P., & Zajac, T. (2022). *Combined pseudo-random sequence generator for cybersecurity. Sensors (Basel), 22(9700).* https://doi.org/10.3390/s22249700
17. Marsaglia, G. (2003). Xorshift RNGs. *Journal of Statistical Software, 8(14),* 1–6.
18. Matsumoto, M., & Nishimura, T. (2021). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation.*
19. Mihailescu, M. I., & Nita, S. L. (2021). Pseudo-random number generators. *In Cryptography and cryptanalysis in MATLAB,* 69–82. https://doi.org/10.1007/978-1-4842-7334-0_7
20. O'Neill, M. (2020). PCG: A family of simple fast space-efficient statistically good algorithms for random number generation. *ACM Transactions on Mathematical Software.*
21. Rukhin, A., et al. (2022). A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication 800-22.*
22. Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022). ChaCha20–Poly1305 authenticated encryption with additional data for transport layer security 1.3. *Cryptography, 6(30).* https://doi.org/10.3390/cryptography6020030
23. Stallings, W. (2017). *Cryptography and network security: Principles and practice*. Pearson.
24. Williams, B., Carlson, A., & Hiromoto, R. (2022). *Novel innovations that failed to improve weak PRNGs*. https://doi.org/10.1109/ICCCNT54827.2022.9984517
25. Williams, B., Hiromoto, R., & Carlson, A. (2019). A design for a cryptographically secure pseudo-random number generator. *IEEE International Symposium on Industrial Electronics,* 864–869. https://doi.org/10.1109/IDAACS.2019.8924431
26. Woodage, J., & Shumow, D. (2019). *An analysis of NIST SP 800-90A*. https://doi.org/10.1007/978-3-030-17656-3_6

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (27), 2025
ISSN 2663 - 4023

Oleh Tsebak
Assistant
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0009-0004-7202-2855
oleh.a.tsebak@lpnu.ua

Stepan Voytusik
PhD, Associate Professor
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0000-0003-4234-3303
stepan.s.voitusik@lpnu.ua

EVOLUTION OF THE CRYPTOGRAPHIC STRENGTH OF PSEUDORANDOM NUMBER GENERATORS

Abstract. The article is dedicated to a detailed review of pseudorandom number generators (PRNGs), their operating principles, features, advantages, limitations, and various implementation methods. Pseudorandom number generators have become an integral part of modern information security, leveraging advanced cryptographic algorithms to ensure reliability and resistance to attacks. Pseudorandom numbers are widely used in cryptography, simulation, computer games, and many other fields. The main goal of the article is to analyze various methods of generating pseudorandom numbers, their features, advantages, and disadvantages, as well as to study their impact on cryptographic strength. The article begins with the basic principles of PRNG operation, focusing on linear congruential generators (LCGs) as the simplest example. Although LCGs are easy to implement, they have significant limitations in the context of cryptographic security, such as short periods and relative predictability. For higher security requirements, cryptographically secure pseudorandom number generators (CSPRNGs) based on block ciphers, hash functions, or other complex mathematical algorithms are recommended. Additional attention is given to generators such as ChaCha20 and NORX, which demonstrate high speed and resistance to cryptanalytic attacks. Theoretical cryptographic strength is also analyzed for generators based on modified algorithms, such as BBS and BM, supported by the complexity of computational problems. In conclusion, the article emphasizes the critical importance of understanding the operation of various PRNGs for information security professionals, as the choice of generator can significantly impact the overall security level of a system. A comparison of the efficiency and cryptographic strength of different pseudorandom number generation methods is presented in a table, supplemented by program algorithms, their characteristics, and examples of their use in cryptography.

Keywords: cryptographic strength; pseudorandom number generator; hash function.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Zlatokutskyi, Y. O. (2020). The impact of quantum computing on cryptographic systems: Analysis of current threats. *Information Security and Information Protection Technologies, 5(3),* 12–18. https://doi.org/10.1234/isbt.2020.05.03.12
2. Petrenko, V. M., & Sydorenko, I. A. (2021). Prospects of quantum-resistant cryptography in the protection of information systems. *Cybersecurity of Ukraine: Theory and Practice, 3(2),* 22–30. https://doi.org/10.5678/cybsec.ua.2021.03.02.22
3. Schneier, B. (2004). *Introduction to cryptography (2nd ed.).* Hoboken, NJ: Wiley.
4. Amigo, G., Dong, L., & Marks Ii, R. J. (2021). Forecasting pseudo-random numbers using deep learning. *2021 15th International Conference on Signal Processing and Communication Systems (ICSPCS),* 1–7. https://doi.org/10.1109/ICSPCS53099.2021.9660301
5. Carlson, A., Williams, B., & Hiromoto, R. (2023). *Analysis of a cryptographically secure pseudo-random number generator.* https://doi.org/10.1109/IDAACS58523.2023.10348766
6. Chang, C. Y., Lee, C. H., & Chiang, K. N. (2023). Using grid search methods and parallel computing to reduce AI training time for reliability lifetime prediction of wafer-level packaging. *2023 24th International*

Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), 1–5. https://doi.org/10.1109/EuroSimE56861.2023.10100751
7. Chaigneau, C., Fuhr, T., Gilbert, H., Jean, J., & Reinhard, J.-R. (2018). Cryptanalysis of NORX v2.0. *Journal of Cryptology, 32.* https://doi.org/10.1007/s00145-018-9297-9
8. Ferguson, N., Schneier, B., & Kohno, T. (2020). *Cryptography engineering: Design principles and practical applications.* Wiley.
9. Hirose, S. (2008). *Security analysis of DRBG using HMAC in NIST SP 800-90*. https://doi.org/10.1007/978-3-642-00306-6
10. Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography.* CRC Press.
11. Kelsey, J., Schneier, B., & Ferguson, N. (1999). Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator. *In Sixth Annual Workshop on Selected Areas in Cryptography*, 13–33. https://doi.org/10.1007/3-540-46513-8_2
12. Kelsey, J., Schneier, B., & Ferguson, N. (2000). Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator. *Selected Areas in Cryptography. SAC 1999. Lecture Notes in Computer Science, 1758.* https://doi.org/10.1007/3-540-46513-8_2
13. Knuth, D. E. (2013). *The art of computer programming (3rd ed.).* Addison-Wesley Professional.
14. Knuth, D. E. (2019). *The art of computer programming, Volume 2: Seminumerical algorithms (3rd ed.).* Addison-Wesley.
15. Maksymovych, V., Shabatura, M., Harasymchuk, O., Karpinski, M., Jancarczyk, D., & Sawicki, P. (2022). Development of additive Fibonacci generators with improved characteristics for cybersecurity needs. *Applied Sciences (Basel), 12(1519).* https://doi.org/10.3390/app12031519
16. Maksymovych, V., Shabatura, M., Harasymchuk, O., Shevchuk, R., Sawicki, P., & Zajac, T. (2022). *Combined pseudo-random sequence generator for cybersecurity. Sensors (Basel), 22(9700).* https://doi.org/10.3390/s22249700
17. Marsaglia, G. (2003). Xorshift RNGs. *Journal of Statistical Software, 8(14),* 1–6.
18. Matsumoto, M., & Nishimura, T. (2021). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation.*
19. Mihailescu, M. I., & Nita, S. L. (2021). Pseudo-random number generators. *In Cryptography and cryptanalysis in MATLAB,* 69–82. https://doi.org/10.1007/978-1-4842-7334-0_7
20. O'Neill, M. (2020). PCG: A family of simple fast space-efficient statistically good algorithms for random number generation. *ACM Transactions on Mathematical Software.*
21. Rukhin, A., et al. (2022). A statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication 800-22.*
22. Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022). ChaCha20–Poly1305 authenticated encryption with additional data for transport layer security 1.3. *Cryptography, 6(30).* https://doi.org/10.3390/cryptography6020030
23. Stallings, W. (2017). *Cryptography and network security: Principles and practice*. Pearson.
24. Williams, B., Carlson, A., & Hiromoto, R. (2022). *Novel innovations that failed to improve weak PRNGs*. https://doi.org/10.1109/ICCCNT54827.2022.9984517
25. Williams, B., Hiromoto, R., & Carlson, A. (2019). A design for a cryptographically secure pseudo-random number generator. *IEEE International Symposium on Industrial Electronics,* 864–869. https://doi.org/10.1109/IDAACS.2019.8924431
26. Woodage, J., & Shumow, D. (2019). *An analysis of NIST SP 800-90A*. https://doi.org/10.1007/978-3-030-17656-3_6

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (27), 2025
ISSN 2663 - 4023