Міністерство освіти і науки України
Харківського національного університету імені В.Н. Каразіна
Навчально-наукового інституту комп'ютерних наук та штучного інтелекту
Спеціальність 125 «Кібербезпека та захист інформації»
Освітня програма «Кібербезпека»

В.о. зав. кафедрою КІСМТ
Марина ЄСІНА
«Допущено до захисту»

“ “ _____2025р.

Пояснювальна записка
до кваліфікаційної роботи бакалавра
на тему: «Перспективні методи та засоби генерування, аналізу та оцінки випадкових і псевдовипадкових послідовностей»

оцінка “_____”
Голова ЕК:
Мичуда Л. З.

Керівник: к.т.н., доцент Єсіна М. В.
Рецензент: к.т.н., професор Качко О. Г.
Виконавець: студент групи КБ-41
Горбенко Дмитро Юрійович

Харків 2025

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра містить 60 сторінок, 28 рисунків, 13 таблиць, 6 додатків і 70 посилань на джерела.

Мета роботи полягає в обґрунтуванні, дослідженні та практичній реалізації перспективних методів генерування ВП і ПВП, їх аналізу та оцінки, а також порівняння для потреб кібербезпеки.

Об’єкт дослідження – методи та засоби генерування, аналізу та оцінки ВП та ПВП.

Предмет дослідження – алгоритми генерації на основі ДНК, статистичного і стохастичного тестування та кількісного порівняння послідовностей, зокрема методи NIST STS, DIEHARD, а також k-мер та MinHash-аналіз.

Основними методами досліджень є статистичне тестування, стохастичний аналіз, експериментальні моделювання й програмна реалізація алгоритмів.

У роботі досліджено: статистичні й ентропійні властивості «сирих» ДНК-послідовностей та послідовностей після екстракції випадковості; працездатність розробленого генератора ВП/ПВП на основі ДНК із криптографічним посиленням шифром «Калина» в режимі CTR (екстракцією); комплекс методів оцінки згенерованих послідовностей, що поєднує стохастичні (ентропійні) метрики та статистичні пакети тестів NIST STS і DIEHARD; алгоритми порівняння згенерованих на основі ДНК послідовностей методами k-мерів та MinHash.

Результати роботи можуть бути використані під час сертифікаційних випробувань генераторів випадкових чисел, у модулях криптографічного захисту інформації для генерації ВП, і відповідно, для підвищення стійкості сучасних криптосистем.

Ключові слова: ВИПАДКОВА ПОСЛІДОВНІСТЬ, ПСЕВДОВИПАДКОВА ПОСЛІДОВНІСТЬ, ЕНТРОПІЯ, PTRNG, NPTRNG, ДНК, NIST STS, DIEHARD, k-МЕР, MINHASH, ЕКСТРАКТОР, КІБЕРБЕЗПЕКА.

ABSTRACT

Bachelor’s qualification work: 60 pages, 28 figures, 13 tables, 6 appendices and 70 references.

The aim of the work is to substantiate, investigate and practically implement promising methods for generating random sequences (RS) and pseudorandom sequences (PRS), their analysis and evaluation, as well as comparison for cybersecurity needs.

Object of research – methods and tools for generating, analyzing and evaluating RS and PRS.

Subject of research – DNA-based generation algorithms, statistical and stochastic testing and quantitative comparison of sequences, in particular the NIST STS and DIEHARD methods, as well as k-mer and MinHash analysis.

The main research methods are statistical testing, stochastic analysis, experimental modeling and software implementation of algorithms.

This study investigates: the statistical and entropic properties of “raw” DNA sequences and sequences after randomness extraction; the operability of the developed DNA-based RS/PRS generator with cryptographic strengthening by the “Kalyna” cipher in CTR mode (extraction); a set of evaluation methods for the generated sequences that combines stochastic (entropic) metrics and the statistical test suites NIST STS and DIEHARD; algorithms for comparing DNA-based generated sequences by k-mer and MinHash methods.

The results of the work can be used during certification tests of random number generators, in information-protection cryptographic modules for RS generation and, accordingly, to increase the resilience of modern cryptosystems.

Keywords: RANDOM SEQUENCE, PSEUDORANDOM SEQUENCE, ENTROPY, PTRNG, NPTRNG, DNA, NIST STS, DIEHARD, k-MER, MINHASH, EXTRACTOR, CYBERSECURITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

APD – Avalanche Photodiode

ASE – Amplified Spontaneous Emission

CTR – Counter Mode

DDBJ – DNA Data Bank of Japan

DNA – Deoxyribonucleic Acid

DRBG – Deterministic Random Bit Generator

FIRO – Fibonacci Ring Oscillator

GARO – Galois Ring Oscillator

GM – Geiger-Müller Counter

LFSR – Linear Feedback Shift Register

NPTRNG – Non-Physical True Random Number Generator

PMT – Photomultiplier Tube

PRNG – Pseudorandom Number Generator

PTRNG – Physical True Random Number Generator

QRNG – Quantum Random Number Generator

RC – Resistor-Capacitor

RNG – Random Number Generator

RO – Ring Oscillator

SpRS – Spontaneous Raman Scattering

SRS – Stimulated Raman Scattering

TFF – Toggle Flip-Flop

UWB – Ultra-Wideband

АЦП – Аналого-Цифровий Перетворювач

ВП – Випадкова Послідовність

ГВЧ – Генератор Випадкових Чисел

ДНК – Дезоксирибонуклеїнова Кислота

ДШ – Джерело Шуму

ПВП – Псевдовипадкова Послідовність

ВСТУП

Сучасний етап розвитку інформаційних технологій супроводжується стрімким підвищенням обчислювальної продуктивності та постійним удосконаленням методів аналізу даних. Такий прогрес, із одного боку, відкриває нові можливості, а з іншого – ставить під загрозу надійність класичних механізмів захисту інформації. Критично важливим чинником безпеки в нових умовах стає забезпечення випадковості, адже саме ВП і ПВП слугують фундаментом стійких ключів шифрування, електронних підписів, протоколів обміну, автентифікації та цілої низки інших криптографічних процедур та перетворень. Ретельний аналіз міжнародних і вітчизняних науково-технічних досліджень підтверджує актуальність пошуку перспективних методів генерування, аналізу та оцінки ВП і ПВП, що й визначило спрямування даної бакалаврської роботи.

Важливість дослідження обумовлена появою нових потужних апаратно-програмних засобів атак: від спеціалізованих графічних і FPGA-кластерів до первинних, але стрімко прогресуючих квантових обчислювачів, здатних реалізовувати нетрадиційні алгоритми криптоаналізу. Для протидії таким загрозам необхідно гарантувати високу ентропію ключового матеріалу та мінімізувати будь-яку передбачуваність у вихідних даних генератора. Отже, поглиблене вивчення джерел істинної випадковості, вдосконалення алгоритмічних генераторів, а також апробація нетрадиційних підходів – наприклад, використання ДНК-послідовностей як природного носія випадкової інформації – стають критично необхідними для довгострокової безпеки сучасних інформаційних систем.

Метою роботи стало наукове обґрунтування, експериментальне дослідження та практична реалізація методів і засобів генерування ВП та ПВП на основі ДНК-послідовностей; розробка комплексної статистичної й стохастичної методики оцінки їхньої якості; а також побудова алгоритмів k-мерного та MinHash-аналізу для кількісного порівняння послідовностей

подібного походження. При цьому увагу також приділено мінімізації обчислювальних витрат, аби забезпечити придатність запропонованих рішень для практичних та ресурсобюджетних середовищ.

Результати, отримані під час виконання роботи, можуть бути безпосередньо впроваджені у сфері кібербезпеки: від апаратних модулів генерації ентропії та програмних бібліотек до систем тестування, сертифікації та моніторингу якості ВП у криптографічних продуктах.

Дослідження логічно пов’язане з глобальною тенденцією до створення стійких до квантових атак криптосистем і ґрунтується на здобутках сучасної теорії інформації, криптографії та біоінформатики. Воно узагальнює та розвиває ідеї світових і національних стандартів (NIST SP 800-90B, FIPS 140-3, ДСТУ 7624:2014 тощо) і пропонує нові методичні підходи, що розширюють теоретичну й практичну базу знань щодо оцінки випадковості.

Таким чином, виконана бакалаврська робота вносить вагомий внесок у розвиток технологій генерування й контролю ВП і ПВП, формуючи підґрунтя для подальших досліджень і розробок у цій критично важливій для кібербезпеки області.

1 КЛАСИФІКАЦІЯ ТА ОПИС ДШ, ЩО МОЖУТЬ ЗАСТОСОВУВАТИСЯ ДЛЯ ГЕНЕРАЦІЇ ВП НА ОСНОВІ PTRNG

1.1 Загальна класифікація

Усі існуючі конструкції ДШ можна поділити на два класи[1]:

- Фізичні ДШ (PTRNG) – базуються на непередбачуваних явищах.
- Нефізичні ДШ (NPTRNG) – ґрунтуються на передбачуваних явищах із певною ентропією.

Прикладами нефізичних ДШ є:

- Переривання апаратних пристроїв (наприклад, мережевої карти);
- Події зчитування/запису на носіях інформації;
- Взаємодія з користувачем.

Джерелом ентропії для нефізичних ДШ слугують часові мітки високої роздільної здатності, молодші біти яких є непередбачуваними.

Фізичні ДШ поділяються на [2, 3]:

- ДШ на основі шуму – використовують непередбачувані шумові процеси;
- ДШ на основі хаосу – залежать від хаотичної поведінки складних систем;
- ДШ на основі вільних осциляторів – базуються на непередбачуваності цифрових перехідних процесів (застосовуються у PTRNG для персональних пристроїв);
- ДШ на квантових ефектах – отримують ентропію за рахунок квантових явищ.

Детальний розгляд кожного класу подано в наступних підрозділах.

1.2 ДШ на основі шуму

ДШ цього класу використовують електричний шум квантової природи, хоча колективна поведінка частинок розмиває його випадковість. Такі RNG називають «квантовими», бо їхня випадковість зумовлена мікроскопічними

процесами, керованими законами квантової механіки. Шум також виникає через взаємодію частинок і охоплює квантовий та класичний рівні.

Рух електронів, що генерують шум, корелює через міжчастинкові взаємодії, знижуючи рівень випадковості [4]. Автокореляція навіть у межах кількох відсотків може суттєво вплинути на точність моделювань при критичних розрахунках. Оскільки шум неможливо «перезапустити», послідовні вимірювання можуть бути корельованими.

Основні фізичні ефекти, що застосовуються:

- Шум Джонсона [5]: випадкові флуктуації напруги в резистивних матеріалах, викликані тепловим рухом квантованого заряду (носіїв). Ентропія достатня для генерації послідовностей, хоча напруга не абсолютно випадкова;
- Шум Зенера: імпульси при тунелюванні носіїв через квантові бар'єри у стабілітронах. При низькому струмі формується шум високої випадковості; лавинний ефект підсилює сигнал і знижує чутливість до електромагнітних впливів;
- Інші джерела шуму [6, 7]: зворотний пробій у біполярних транзисторах, фазовий шум лазерів, хаотичні коливання.

Проблеми RNG на основі шуму:

- Низька амплітуда шуму потребує підсилення перед перетворенням у цифровий сигнал, що може вносити спотворення через нелінійність підсилювача та обмежену смугу пропускання;
- Електромагнітні завади від цифрової логіки можуть спричиняти синхронізацію генераторів, знижуючи ентропію.

PTRNG на основі шуму працює за принципом порівняння випадкової аналогової напруги з пороговим значенням: якщо вище порогу – формується біт «1», інакше – «0» (рис. 1.1). Точне налаштування порогу є проблематичним через вплив температури та зміни напруги живлення.

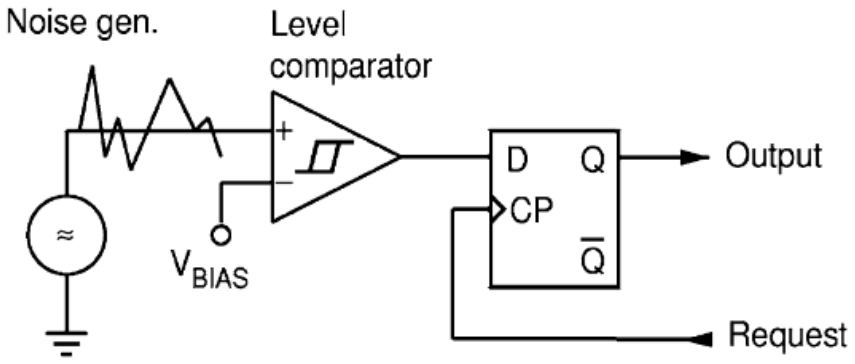

Рисунок 1.1 – Загальна конструкція PTRNG на основі шуму

Для покращення випадковості RNG на основі шуму було запропоновано декілька схем, спрямованих на зменшення кореляції між згенерованими бітами та покращення рівномірності їх розподілу.

Генератор Багіні–Букчі [8], зображений на рис. 1.2: періодична дискретизація напруги від джерела шуму, компаратор та тригер Т-типу (TFF). Завдяки цьому механізму вихідний сигнал має тенденцію до рівномірного розподілу «0» і «1» у довгостроковій перспективі. Однак схема може страждати від ефекту пам’яті – напруга, що подається на компаратор, може залежати від попередніх значень через паразитні ємності.

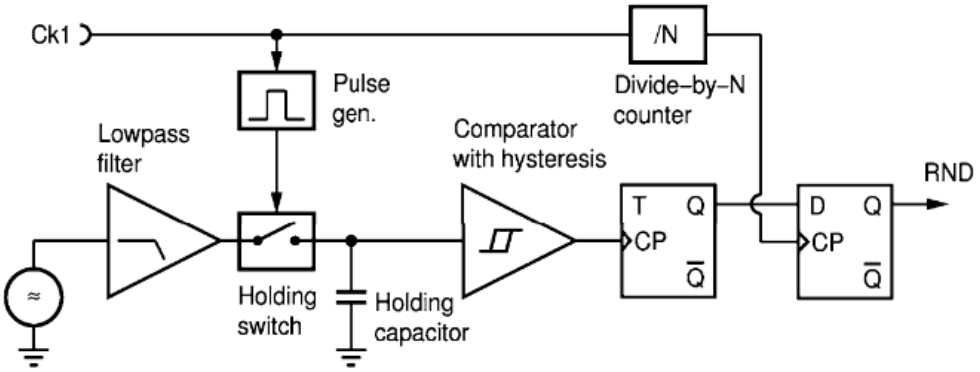

Рисунок 1.2 – Генератор Багіні–Букчі

Метод “сумування за часом” (рис. 1.3): імпульси компаратора накопичуються в лічильнику за модулем 2 (TFF). Під час запиту (Request) лічильник фіксує значення, що зменшує кореляцію між сусідніми бітами та забезпечує швидшу генерацію [9, 10].

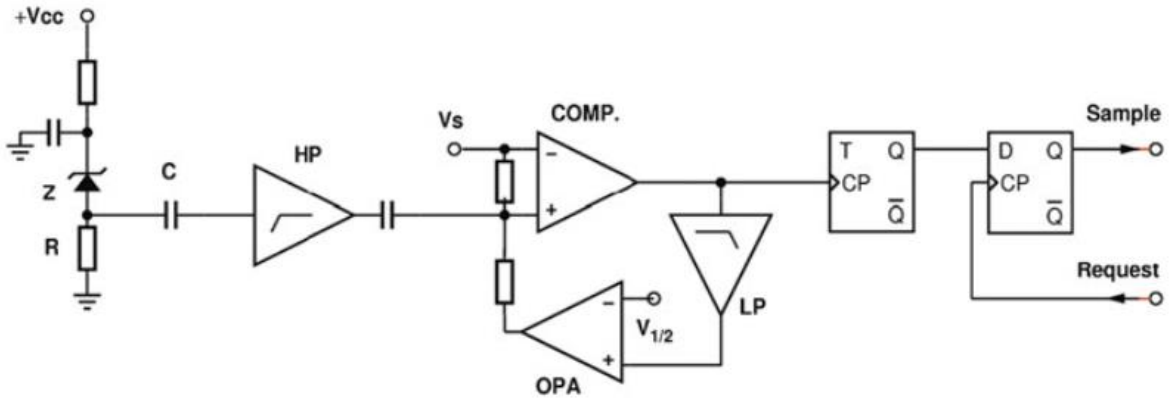

Рисунок 1.3 – PTRNG на основі шуму без зміщення розподілу ймовірностей

Крім апаратних удосконалень, застосовуються методи післяобробки, що допомагають мінімізувати можливі відхилення від ідеальної випадковості:

- XOR-декореляція: побітова операція XOR над кількома бітами для зменшення кореляцій;
- Усунення зміщень фон Неймана [11]: видалення надмірних повторюваних патернів;
- Гешування криптографічними функціями (SHA, Keccak): вирівнює розподіл бітів;
- Лінійні зсувні регістри (LFSR): покращують статистичну випадковість без додавання ентропії.

Залежно від застосування може бути необхідним комбінування декількох методів для досягнення потрібного рівня випадковості.

Підсумовуючи, повна математична верифікація випадковості RNG на основі шуму складна через недосконалу ізоляцію фізичних процесів і вплив зовнішніх факторів.

1.3 ДШ на основі хаосу

Серед ефективних хаотичних систем (оптичних, електричних та оптико-електричних) особливо поширені лазерні PTRNG, які забезпечують високу швидкість генерації, компактність і низьке енергоспоживання. Вони реалізовуються через зворотний зв’язок у лазерному резонаторі (рис. 1.4), що викликає хаотичну флуктуацію інтенсивності світла; фотодіод перетворює її на

електричний сигнал, який дискретизується АЦП та обробляється математично, дозволяючи інтегрувати генератор у мікросхеми та повністю оптичні обчислювальні середовища [12].

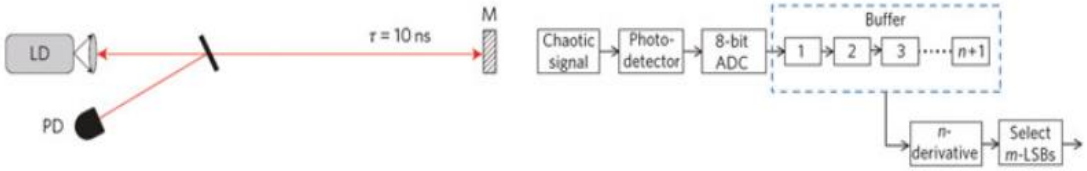

Рисунок 1.4 – Приклад лазерної PTRNG на основі хаосу

Генератор на основі надширокосмугового (UWB) хаотичного лазера (рис. 1.5) складається з двох лазерів із розподіленим зворотним зв'язком [13]: головний збурює підлеглий, розширюючи його смугу пропускання. Інтенсивність світла порівнюється з порогом оптичним компаратором, що формує випадкові біти.

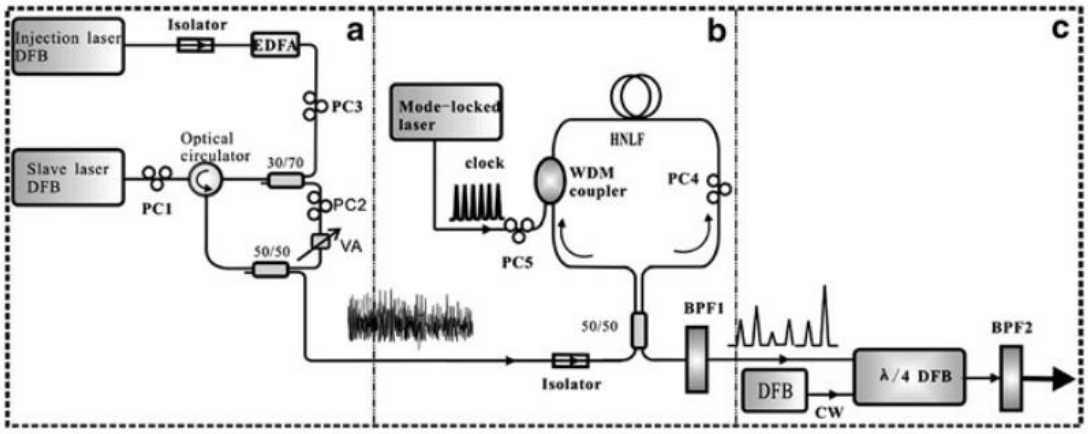

Рисунок 1.5 – Використання хаотичного лазера для PTRNG

Для підтвердження хаотичної поведінки таких генераторів застосовують показники Ляпунова [14] та експериментальні моделі [15]. Проте хаотичні RNG детерміновані диференційними рівняннями з обмеженою інформацією й не генерують нову ентропію довгостроково: як тільки інформація вичерпується, система припиняє видавати нову ентропію. У реальних умовах це обмеження частково компенсується випадковими квантовими або статистичними ефектами.

Проте ці ефекти не є основою хаотичних RNG, оскільки їхній внесок у випадковість є незначним. Єдиним джерелом фундаментальної невизначеності залишається квантова випадковість.

1.4 ДШ на вільних осциляторах

Логічний інвертор із підключеним виходом до входу (рис. 1.6) працює як генератор на вільних осциляторах. Теоретично, такий контур не має визначеного стану, але через затримку поширення сигналу він починає коливатися. Частота коливань залежить від параметрів схеми, включаючи напругу живлення, температуру та внутрішній шум, що робить її придатною для генерації випадкових чисел.

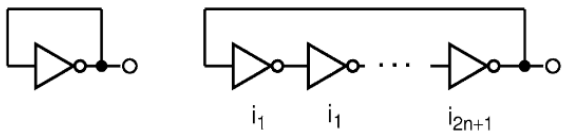

Рисунок 1.6 – PTRNG на вільних осциляторах

Основний механізм генерації випадковості базується на взаємодії швидкого та повільного осциляторів, де вихід першого дискретизується другим. Випадкове тремтіння фази та частоти забезпечує необхідний рівень ентропії. Проте при недостатньому тремтінні або синхронізації осциляторів генератор може виробляти передбачувані патерни, що знижує якість випадковості.

Головні проблеми RNG на основі вільних осциляторів:

- Висока чутливість до зовнішніх впливів (температура, електромагнітне випромінювання).
- Схильність до синхронізації сусідніх осциляторів, що знижує рівень ентропії.
- Необхідність пост-обробки для усунення кореляцій у вихідних даних.

Попри це, RNG на вільних осциляторах залишаються популярним вибором завдяки простоті реалізації та можливості інтеграції у мікросхеми [16]. Одним із вдосконалень є використання схем FIRO (кільцевий осцилятор Фібоначчі) та GARO (кільцевий осцилятор Галуа) [17], що поєднують вільні осцилятори з

регістрами зсуву (рис. 1.7). Ці конструкції покращують випадковість, зменшуючи кореляції, проте залишаються чутливими до параметрів реалізації.

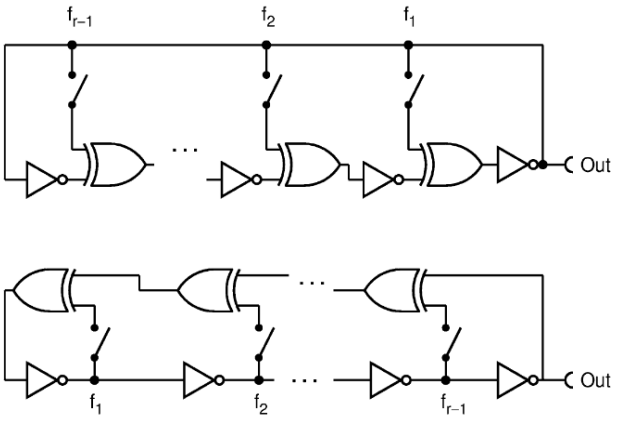

Рисунок 1.7 – Схема GARO і FIRO

RNG на основі вільних осциляторів – це економічне, але не ідеальне рішення через труднощі з доведенням випадковості та необхідність ретельного налаштування для забезпечення достатнього рівня ентропії.

1.5 ДШ на основі радіоактивного розпаду

Радіоактивний розпад став одним із перших квантових явищ, використаних для генерації випадкових чисел [18]. RNG цього типу використовують детектори (найчастіше GM-Трубки) та радіоактивні джерела випромінювання (α, β, γ). Випадковість подій розпаду підпорядковується розподілу Пуассона, а імпульси, що реєструються детектором, можуть бути використані для генерації випадкових чисел.

Методи отримання випадкових чисел:

- Швидка синхронізація [19] – частота синхронізації перевищує середню швидкість виявлення подій.
- Повільний годинник [18] – підрахунок імпульсів відбувається частіше, ніж тактовий цикл (рис 1.8).

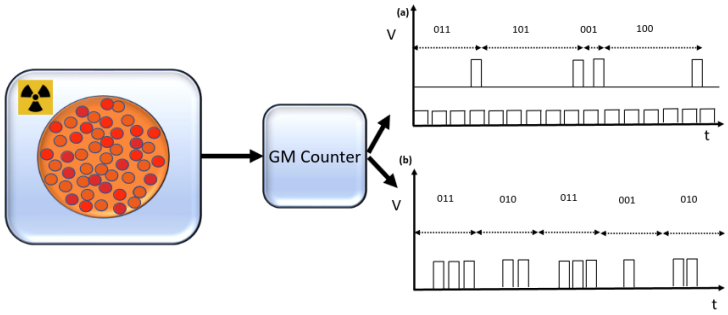

Рисунок 1.8 – Принцип дії методу повільного годинника

Реалізація RNG передбачає вимірювання часу між послідовними імпульсами, що дозволяє отримати випадкові біти. Для покращення рівномірності розподілу використовується паритет підрахунку або модульне додавання різних підрахунків, що мінімізує зсуви.

Сучасні реалізації використовують напівпровідникові детектори замість GM-Трубок, що спрощує конструкцію та зменшує енергоспоживання. Додаткове посилення сигналу компенсує їх нижчу чутливість. Одним із методів вирівнювання розподілу є застосування RC-ланцюгів для перетворення експоненційного розподілу в рівномірний.

Попри переваги, QRNG на основі радіоактивного розпаду мають обмеження:

- Радіоактивні джерела потребують спеціальних заходів безпеки.
- Детектори можуть втрачати чутливість через радіаційне старіння.
- "Мертвий час" детектора знижує швидкість генерації випадкових чисел.
- Потребують пост-обробки для корекції випадкових бітів.

Через ці фактори генератори на основі радіоактивного розпаду мають обмежене практичне застосування та використовуються лише в спеціалізованих системах.

1.6 ДШ на атомарних системах

Генерація випадкових чисел на основі атомарних систем є перспективним напрямком квантової криптографії. Одним із запропонованих методів є

використання захоплених іонів, хоча такі установки складні та забезпечують низьку швидкість генерації [20, 21].

Інший підхід – QRNG на основі спінового шуму пари лужного металу [22]. Спіновий шум виникає внаслідок квантової невизначеності та взаємодії атомів. У такій системі шум досліджується оптично за допомогою лазерного випромінювання, яке перетворюється на поляризацію світла, що дозволяє генерувати випадкові числа. Для ефективності необхідно мінімізувати вплив фонових шумів.

Окрім цього, використання твердотільних систем може значно підвищити швидкість генерації випадкових бітів, роблячи їх більш придатними для практичного застосування.

1.7 ДШ на однофотонних детекторах

Генерація випадкових чисел у цьому класі ДШ ґрунтується на квантовому принципі суперпозиції станів та колапсі хвильової функції при вимірюванні. Однофотонні детектори фіксують випадкові значення стану фотона після проходження крізь розсіювач променя або поляризаційний роздільник. Після вимірювання фотон набуває одного з можливих станів, що дозволяє отримати випадковий біт.

Реалізації QRNG (рис. 1.9) базуються на:

- Оптичних системах – використання роздільників променя, фотопомножувачів, лавинних фотодіодів [23-25].
- Хмарних квантових комп'ютерах – генерування випадкових чисел через вимірювання кубітів у суперпозиції [26].

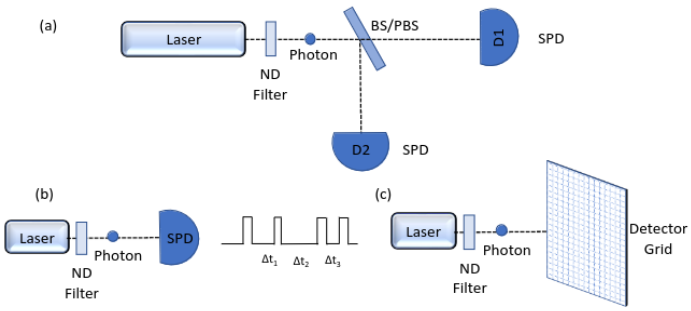

Рисунок 1.9 – QRNG на основі часу проходження фотонів

Головні обмеження QRNG на однофотонних детекторах [23, 27]:

- Часова неактивність детектора після виявлення фотона спричиняє кореляцію між бітами.
- Декілька одночасно зареєстрованих фотонів можуть вплинути на випадковість.
- Недосконаліть детекторів може призводити до хибних спрацьовувань.

Для підвищення швидкості генерації використовують кілька шляхів фотонів, що дозволяє збільшити кількість отриманих бітів, проте ускладнює реалізацію генератора [28].

1.8 Висновки до розділу

Табл. 1.1 в якості висновків містить порівняльний аналіз основних типів фізично справжніх ДШ, що застосовуються для генерування ключових даних та загальних параметрів для існуючих класичних та квантовостійких криптоперетворень та криптопротоколів. У таблиці наведено ключові переваги, недоліки та специфічні особливості кожного ДШ. Опис інших фізично справжніх ДШ наведено у додатку А.

Таблиця 1.1 – Переваги та недоліки ДШ

Тип	Переваги	Недоліки	Особливості
На основі шуму	• Фундаментальна квантова випадковість • Різні джерела шуму (Джонсон, Зенер)	• Автокореляція через взаємодію носіїв • Низька амплітуда потребує підсилення (спотворення) • Вразливі до електромагнітних перешкод	• Використовують електричні шуми • Основні фізичні ефекти: шум Джонсона, шум Зенера
На основі хаосу	• Різноманітні фізичні реалізації (оптичні, електричні, оптико-електричні)	• Складно підтвердити хаотичність довгостроково	• Зворотний зв’язок у лазерному резонаторі • Оцінка хаотичності через Ляпуновські показники

Продовження таблиці 1.1

Тип	Переваги	Недоліки	Особливості
	• Компактність і висока швидкість лазерних схем	Обмежена генерована ентропія без квантових домішок	
На вільних осциляторах	• Простота конструкцій кільцевих інверторів • Легко інтегруються в ПЛІС та мікросхеми	• Чутливі до температури, напруги та синхронізації • Пам’ять через паразитні ємності (memory effect)	• Схеми FIRO і GARO з регістрами зсуву • Взаємодія швидкого і повільного осциляторів
На основі радіоактивного розпаду	• Випадковість Poisson-процесу з доказовою квантовою основою • Історично перші QRNG із багатьма дослідженнями	• Потребує радіаційної безпеки • «Мертвий час» детектора знижує швидкість генерації	• Детектори: GM-трубки або напівпровідникові сенсори • Методи: швидка синхронізація, повільний годинник
Атомарні системи	• Фундаментальна квантова ентропія іонних та спінових процесів • Перспективи для квантових технологій	• Складна експериментальна установка • Дуже низька швидкість генерації	• Джерело: захоплені іони • Спіновий шум лужних металів
На основі однофотонних детекторів	• Квантова випадковість через колапс фотонів • Можуть досягати високих швидкостей	• Обмеження «мертвого часу» та темних імпульсів • Кореляції від мультифотонних подій	• Розділювач променя/поляризації + APD/PMT • Мультиканальні схеми для збільшення пропускної здатності

2 МЕТОДИ ТА ЗАСОБИ АНАЛІЗУ ТА ОЦІНКИ ВЛАСТИВОСТЕЙ ВП

2.1 Статистичні методи

Перший крок до стандартизації статистичних тестів у США було зроблено в 1994 році шляхом ухвалення національного стандарту «Вимоги безпеки до криптографічних модулів». Стандарти серії FIPS 140 містять вимоги до криптографічних модулів, зокрема їхні три версії: FIPS 140-1 [29], FIPS 140-2 [30] та FIPS 140-3 [31]. Проте ці стандарти мають переважно технологічну спрямованість і розв’язують завдання статистичного контролю в криптографічних модулях, що генерують ПВП, але є малопридатними для дослідження статистичних властивостей ГВЧ.

Набір статистичних тестів DIEHARD [32], створений Джорджем Марсарією у 1995 році, призначений для оцінки якості випадкових послідовностей. Спочатку він поширювався на компакт-дисках і досі вважається одним із найжорсткіших тестових наборів.

У 1999 році в межах проєкту AES (Advanced Encryption Standard) фахівці NIST розробили NIST STS (NIST Statistical Test Suite), запропонувавши методику статистичного тестування ГВЧ (ГПВЧ), орієнтовану на потреби криптографічного захисту інформації. Багато експертів вважають цей набір найбільш відповідним сучасним вимогам тестування [33, 34].

Роберт Г. Браун із Університету Дюка розширив DIEHARD до набору DIEHARDER [35], який дозволяє приймати однозначні рішення щодо непридатності генератора (наприклад, із рівнем відмови 0,0001%) замість класичних 1% чи 5%. DIEHARDER містить оновлені тести DIEHARD, NIST STS і нові тести, розроблені самим Брауном. Набір постійно доповнюється і зрештою перевершить поточну кількість тестів [36].

2.1.1 Критерії прийняття рішень щодо випадковості послідовності

Найбільш поширеним на практиці підходом до визначення псевдовипадковості є евристичний підхід. При цьому підході псевдовипадковий генератор розглядається як програма (алгоритм), яка породжує бітову

послідовність $s = s_0, s_1, .., s_{n-1}$ скінченої довжини n, що проходить деякі особливі статистичні тести. Таким чином властивості випадкової або псевдовипадкової послідовності можуть бути охарактеризовані та описані в імовірнісному значенні. Жоден набір тестів не може гарантувати абсолютного визначення випадковості, тому їх результати слід інтерпретувати з обережністю.

В основі статистичних тестів лежить перевірка нульової гіпотези H_0, що послідовність випадкова, проти альтернативної гіпотези H_A, яка припускає її невипадковість. Для кожного тесту обчислюється відповідна статистика, яка порівнюється з критичним значенням. Якщо тестова статистика перевищує критичне значення, нульова гіпотеза відхиляється, і послідовність вважається невипадковою, а якщо ні – гіпотеза випадковості приймається.

При цьому можливі дві основні помилки. Помилка 1-го роду α виникає, коли випадкова послідовність помилково не проходить тест і відкидається як невипадкова. Це означає, що «добрий» генератор відбраковується через надмірно жорсткі критерії тестування. Помилка 2-го роду β має місце, коли невипадкова послідовність хибно приймається за випадкову, що є значно критичнішим, адже «поганий» генератор залишається невиявленим. Для підвищення надійності тестування необхідно правильно обирати рівень значущості тесту α та зменшувати ймовірність помилки 2-го роду β.

Існують три основні підходи до прийняття рішення щодо випадковості послідовності.

Нехай дана двійкова послідовність $S = \{s_1, s_2, ..., s_n\}$, $s_i \in \{0,1\}$ довжиною n біт. Необхідно прийняти рішення, проходить ця послідовність статистичний тест на випадковість чи ні. Можливі наступні підходи до вирішення цієї задачі.

Критерій прийняття рішення на основі встановлення граничного рівня. Базується на встановленні граничного рівня, де тестова статистика $c(S)$ порівнюється з пороговим значенням $c_{пор}(S)$. Якщо $c(S) < c_{пор}(S)$, послідовність не проходить тест і вважається невипадковою.

Наприклад, тест Лемпеля-Зіва аналізує складність послідовності і, якщо вона нижча за $n/\log_2 n$, припускається наявність регулярних структур. Однак цей метод має обмежену надійність, оскільки встановлення одного фіксованого критерію може не враховувати специфічні особливості різних послідовностей.

Критерій прийняття рішень на основі встановлення фіксованого довірчого інтервалу. У цьому випадку послідовність не проходить тест, якщо значення тестової статистики $c(S)$ виходить за межі інтервалу, що відповідає заданому рівню значущості α.

Наприклад, у частотному тесті очікується приблизно рівномірний розподіл нулів та одиниць у послідовності. Якщо частка одиниць значно відхиляється від 50% і виходить за межі довірчого інтервалу, послідовність вважається невипадковою. Цей метод є надійнішим за граничний підхід, оскільки дозволяє враховувати статистичні коливання випадкових величин.

Третій підхід побудови критерію прийняття рішення спирається на обрахування для статистики тесту $c(S)$ відповідного значення ймовірності $P-value$. Тут статистика тесту розглядаються як реалізація випадкової величини, яка підкоряється відомому закону розподілу. Статистика тесту будується таким чином, щоб її менші значення вказували на будь-який дефект випадковості послідовності. Значення ймовірності $P-value$ є ймовірність того, що статистика тесту прийме значення, більше за значення, що спостерігається при випробуванні послідовності, у передбаченні її випадковості. Якщо $P-value$ <0.01, тобто лише одна з 100 випадкових послідовностей мала б таке відхилення, вважається, що послідовність не випадкова. Аналогічно, при $P-value$<0.001, коли лише одна з 1000 випадкових послідовностей мала б подібне відхилення, ймовірність випадковості ще нижча. Основна мета тестів, побудованих за третім підходом, – мінімізація ймовірності помилки 2-го роду, тобто зменшення ризику прийняття послідовності, сформованої «поганим» генератором, за таку, що згенерована «добрим» генератором. Імовірності помилок 1-го α і 2-го β роду взаємопов’язані між собою та з довжиною послідовності n: якщо задано два з цих параметрів, третій визначається автоматично. У практичному застосуванні

зазвичай фіксують розмір послідовності n та рівень значущості тесту α, після чого критичне значення підбирається так, щоб мінімізувати β і, відповідно, знизити ризик хибного прийняття невипадкової послідовності.

2.1.2 Методика тестування NIST STS

Методика тестування NIST STS ґрунтується на перевірці нульової гіпотези (H_0), що тестована послідовність є випадковою. У разі її відхилення приймається альтернативна гіпотеза (H_α), яка вказує на невипадковість. Остаточне рішення щодо випадковості послідовності приймається на основі сукупності результатів усіх тестів.

Процедура тестування окремої двійкової послідовності включає такі етапи:

1) Формулювання гіпотези – припущення, що послідовність є випадковою (H_0).
2) Обчислення тестової статистики для заданої послідовності.
3) Визначення ймовірності P на основі тестової статистики.
4) Прийняття рішення: якщо $P \geq \alpha$, гіпотеза випадковості приймається, інакше – відкидається.

NIST STS містить 16 статистичних тестів, спрямованих на виявлення певних аномалій у послідовності [33, 34]. Загалом, залежно від вхідних параметрів, обчислюється 189 значень ймовірностей, що можна розглядати як результати окремих тестів. Після тестування формується вектор значень ймовірності $P = \{P_1, P_2 ... P_{189}\}$ для оцінки випадковості. Аналіз цих значень дозволяє ідентифікувати конкретні дефекти.

Методика тестування передбачає оцінку випадковості послідовностей, що формуються генератором. Для перевірки фіксується довжина n та формується множина з m двійкових послідовностей. Кожна з них тестується пакетом NIST STS, а результати заносяться до статистичного портрета генератора – матриці розмірності $m \times q$, де q – кількість застосованих тестів.

На основі отриманого статистичного портрета генератора визначається частка послідовностей, що успішно пройшли кожен тест. Для цього встановлюється рівень значущості α, після чого підраховується кількість тестових ймовірностей, що перевищують цей рівень. Результати формують вектор коефіцієнтів $R = \{r_1, r_2, \dots, r_q\}$, елементи якого відображають у відсотках проходження всіх тестів.

Правило 1. Генератор вважається таким, що пройшов тестування за конкретним тестом, якщо його коефіцієнт проходження знаходиться в межах довірчого інтервалу. Межі цього інтервалу визначаються за стандартними статистичними методами.

Значення ймовірностей, отримані після тестування, повинні підкорятися рівноймовірному закону розподілу на інтервалі [0,1]. Для цього будується гістограма частотних розподілів отриманих ймовірностей, після чого рівноймовірність перевіряється за критерієм χ^2 із дев'ятьма ступенями свободи.

Правило 2. Генератор вважається таким, що пройшов тестування за окремим тестом, якщо статистика χ^2 відповідає критерію рівноймовірного розподілу.

Генератор випадкових послідовностей успішно проходить тестування NIST STS, якщо всі коефіцієнти проходження знаходяться в межах довірчих інтервалів і результати тестів відповідають критеріям статистичного аналізу. Це підтверджує відповідність генератора криптографічним вимогам щодо випадковості.

2.1.3 Методика тестування DIEHARD

Набір DIEHARD [32] – це комплекс 15 статистичних тестів, призначених для оцінки якості генератора випадкових чисел. Тести аналізують різні аспекти випадковості, включаючи розподіл чисел, їхні кореляції, закономірності та можливість стиснення. Для коректного тестування рекомендується використовувати послідовності довжиною 10^6 чисел.

Основний критерій оцінки – *P*-значення, яке має бути рівномірно розподілене в діапазоні [0,1], якщо послідовність справді випадкова. Відхилення від цього розподілу свідчать про можливі аномалії у генераторі. Перелік тестів наведено у додатку Б.

План методики тестування:

1) Формування вибірки – створюється множина випадкових послідовностей довжиною 10^6 чисел.

2) Застосування тестів – кожна послідовність проходить усі 15 тестів DIEHARD.

3) Аналіз *P*-значень – оцінюється рівномірний розподіл значень у діапазоні [0,1].

4) Прийняття рішення – якщо всі результати відповідають очікуваному розподілу, генератор визнається таким, що пройшов тестування.

2.2 Стохастичні методи

Традиційно вважається, що статистичне тестування генераторів випадкових чисел (RNG) та відповідність вимогам стандартів [30, 33, 35] є достатнім для підтвердження випадковості. Однак цього недостатньо в умовах можливих атак, зокрема класичних, квантових та атак бічними каналами. Визнано, що окрім статистичного тестування необхідно застосовувати стохастичне тестування, яке базується на аналізі та оцінці початкової ентропії ВП.

2.2.1 Аналіз вимог щодо ДШ та їх ентропії

Основоположні вимоги до джерел шуму та його ентропії було запропоновано в стандарті NIST SP 800-90B. Метою вимог NIST США до джерела ентропії [37, 38] є надання розробнику допомоги в розробці/впровадженні джерела ентропії, яке може надати вихідні дані з постійною кількістю ентропії, а також надати необхідну документацію для перевірки джерела ентропії. Перелік вимог, що висуває стандарт NIST SP800-90B [38] до ДШ у вигляді PTRNG та NPTRNG, наведений у додатку В.

2.2.2 Методи оцінки ентропій ВП, що згенеровані PTRNG

Ентропійні методи оцінки ґрунтуються на використанні узагальненого поняття ентропії Реньї [37, 39]. Нехай X – випадкова змінна, що у найбільш узагальненому виді визначає ентропію Реньї. Для випадкової змінної X ентропію Реньї можливо розрахувати за формулою

$$H_\alpha(X) = \frac{1}{1-\alpha}\log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha, 0 \le \alpha < \infty. \quad (2.1)$$

На практиці значеннями параметра $\alpha \in 1{,}2$ та ∞. При $\alpha \in 1$ із (2.1) отримуємо співвідношення для оцінки ентропії Шеннона, при $\alpha \in 2$ – для оцінки колізійної ентропії, а при $\alpha \epsilon \infty$ – для оцінки мінімальній ентропії.

За визначенням ентропія Реньї $H_\alpha(X)$ залежить тільки від розподілу μ випадкової змінної X, тому будемо використовувати також нотацію $H_\alpha(\mu)$, показуючи залежність $H_\alpha(X)$ як від α, так і від μ.

Розглянемо більш детально кожен випадок і покажемо, що у приватних випадках відповідних α маємо збіг.

Врахуємо, що ентропія Шеннона є границею ентропії Реньї $H_\alpha(X)$ в точці $\alpha = 1$, в результаті отримуємо

$$H_1(x) = \lim_{\alpha\to 1} H_\alpha(x) = \lim_{\alpha\to 1} \frac{1}{1-\alpha}\log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha.$$

Для того щоб отримати класичну формулу ентропії Шеннона з цієї границі скористаємося правилом Лопіталя, згідно з яким для двох дійсних функцій $f(x) = \log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha$ та $g(x) = 1-\alpha$, що мають похідні $f'(x), g'(x)$ в околиці точки δ, має місце вираз

$$\lim_{x\to\delta} \frac{f(x)}{g(x)} = \lim_{x\to\delta} \frac{f'(x)}{g'(x)}$$

У цьому випадку $f(x) = \log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha, g(x) = 1-\alpha$ і $\delta = \alpha = 1$.

Для отримання похідних скористаємося такими формальними правилами взяття похідних складної функції

$$\frac{d}{dx}\log_2 (x) = \frac{1}{x\ln 2}, \frac{d}{dx} a^x = a^x \cdot \ln a.$$

Підставляючи відповідні значення у формулу (2.1), отримуємо:

$$\lim_{\alpha\to 1} H_\alpha(x) = \lim_{\alpha\to 1} \frac{\log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha}{1-\alpha} = \lim_{\alpha\to 1} \frac{\frac{d}{dx}\left(\log_2 \sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha\right)}{\frac{d}{dx}(1-\alpha)} =$$

$$= \lim_{\alpha\to 1} \frac{\left(\sum_{i=1}^{k} (\Pr[X=\omega_i])^\alpha\right)^{-1} \sum_{i=1}^{k} \Pr[X=\omega_i]^\alpha \cdot \ln(\Pr[X=\omega_i])}{-\ln 2}$$

Використовуючи властивість натурального логарифму $\frac{\ln b}{\ln c} = \log_c b$, перетворюємо $\frac{\sum_{i=1}^{k} \ln(\Pr[X=\omega_i])}{-\ln 2}$ і отримаємо наступний вираз:

$$\lim_{\alpha\to 1} H_\alpha(x) = -\sum_{i=1}^{k} \Pr[X=\omega_i] \cdot \log_2(\Pr[X=\omega_i]).$$

Звідки, маємо класичну формулу Шеннона:

$$H_1(X) = H(X) = -\sum_{i=1}^{k} \Pr[X=\omega_i] \log_2(\Pr[X=\omega_i]) \qquad (2.2)$$

Якщо $\Pr[X=\omega_i] = 0$, то, за домовленістю, $\Pr[X=\omega_i]\log_2(\Pr[X=\omega_i]) = 0$. Позначення H зазвичай використовується замість H_1 для ентропії Шеннона. Ентропію Шеннона $H = H_1$ іноді називають загальною ентропією, або просто ентропією через її важливість в теорії інформації [40].

У (2.1) вказано, що мінімальна ентропія є спеціальним випадком для якого $\alpha = \infty$. На основі виразу (2.1) отримаємо аналітичне співвідношення для мінімальної ентропії. Так як $\alpha = \infty$, то на основі обчислення границі скористаємося тим фактом, що для усіх $i = 1,2,\dots k$, $0 \le \Pr[X=\omega_i] \le 1$. При збільшенні α сума $\sum_{i=1}^{k} \Pr[X=\omega_i]^\alpha$ буде наближатися до $\max_i \Pr[X=\omega_i]$.

Позначимо $p_i = \Pr[X=\omega_i]$.

Розглянемо на основі ентропії Реньї альтернативний варіант отримання співвідношення для мінімальної ентропії. Враховуючи прийняте позначення, маємо

$$\lim_{\alpha\to\infty} H_\alpha(x) = \lim_{\alpha\to\infty} \frac{\log_2 \sum_{i=1}^{k} p_i^\alpha}{1-\alpha}$$

Для виділення величини $\max_i p_i^\alpha$ виконаємо перетворення

$$\lim_{\alpha\to\infty} \frac{\log_2 \left(\sum_{i=1}^{k} \left(\left(\frac{p_i}{max_i p_i}\right)^\alpha \cdot max_i p_i^\alpha\right)\right)}{1-\alpha}.$$

Оскільки добуток під логарифмом дає суму двох логарифмів, то справедливо, що

$$\lim_{\alpha\to\infty}\frac{\log_2\sum_{i=1}^{k}\left(\frac{p_i}{max_i p_i}\right)^{\alpha}+\log_2\left(max_i p_i^{\alpha}\right)}{1-\alpha}=$$
$$=\lim_{\alpha\to\infty}\frac{\log_2\sum_{i=1}^{k}\left(\frac{p_i}{max_i p_i}\right)^{\alpha}}{1-\alpha}+\lim_{\alpha\to\infty}\frac{\log_2\left(max_i p_i^{\alpha}\right)}{1-\alpha}.$$

Позначимо, $\beta=\sum_{i=1}^{k}\left(\frac{p_i}{max_i p_i}\right)^{\alpha}$, маємо:

$$\frac{\log_2\beta}{1-\alpha}+\lim_{\alpha\to\infty}\frac{\log_2\left(max_i p_i^{\alpha}\right)}{1-\alpha}.$$

Оскільки, значення p_i пронормоване $\max\limits_i p_i$, то під знаком суми одне $\max\limits_i p_i=1$, а сума всіх інших пронормованих ймовірностей – менше або дорівнює 1.

Тобто, $1<\beta\le k$. Так як $\alpha\to\infty$, то значення $\beta\to\infty$:

$$\frac{\log_2\beta}{\infty}+\lim_{\alpha\to\infty}\frac{\log_2\left(max_i p_i^{\alpha}\right)}{1-\alpha}=\frac{\log_2\beta}{\infty}+\lim_{\alpha\to\infty}\frac{\alpha\log_2\left(max_i p_i\right)}{1-\alpha}.$$

Оскільки β – скінченна величина, то $\frac{\log_2\beta}{\infty}=0$. Тому можна записати для похідних

$$0+\lim_{\alpha\to\infty}\frac{\alpha\log_2\left(max_i p_i\right)}{1-\alpha}=\lim_{\alpha\to\infty}\frac{\frac{d}{d\alpha}\left(\alpha\log_2\left(max_i p_i\right)\right)}{\frac{d}{d\alpha}(1-\alpha)}=-\log_2(max_i p_i)\qquad(2.3)$$

Отримаємо та перевіримо на основі використання виразу для ентропії Реньї вираз для ентропії колізій. Нехай H_2 позначає колізійну ентропію. Нехай також, X та X' – дві незалежні та однаково розподілені випадкові змінні зі значеннями в деякій множині Ω. Тоді із виразу для ентропії Реньї (2.1), при $\alpha=2$, маємо

$$H_2(X)=\frac{1}{1-2}\log_2\left(\sum_{\omega\in\Omega}\left(\Pr\left[X=\omega\right]\right)^2\right)=-\log_2\left(\sum_{\omega\in\Omega}\left(\Pr\left[X=\omega\right]\right)^2\right)\quad(2.4)$$

Практичні дослідження показують, що має місце таке співвідношення між енропією Шеннона, колізійною і мінімальною ентропією [37, 39]:

$$H_{\min}\le H_2\le H_1, H_{\min}\le H_2\le 2H_{\min}.\qquad(2.5)$$

Мінімальна ентропія є найбільш консервативною ентропією. Для прикладу зображені H_1, H_2 та H_{min} для бінарних випадкових змінних. Задаючи значення ймовірностей $\Pr[X = \omega_i]$, по кривим (рис. 2.1) перевіряється, чи виконується практично співвідношення між $H_1, H_2, H_{\min}$ та $2H_{\min}$. Якщо так, то приймається рішення, що послідовність відповідає ентропійним критеріям випадковості. Після цього етапу, для уточнення, рекомендується провести статистичне тестування, використовуючи методи розглянуті у підрозділах вище.

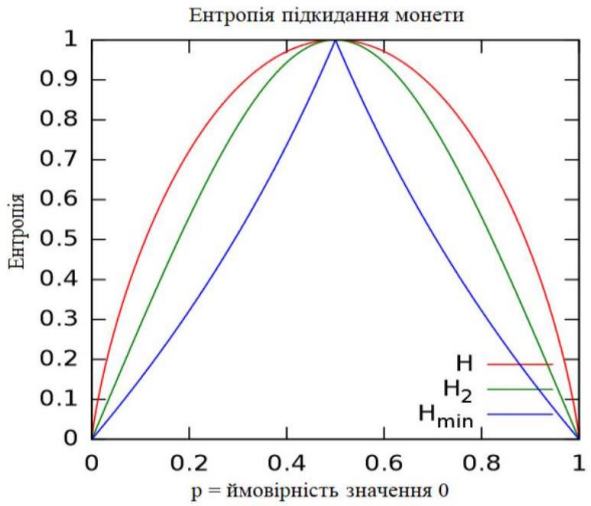

Рисунок 2.1 – Мінімальна ентропія, колізійна ентропія та ентропія Шеннона для випадкових бінарних змінних

2.2.3 Дослідження ентропій Шеннона, ентропії колізій та мінімальної ентропії ДШ згідно 3 NIST 800-90B

Методологічні основи оцінки ентропій ВП та ВЧ, що згенеровані PTRNG фізично справжніми генераторами, наведено в [37, 41-43]. Нижче наводяться методики оцінки ентропій ВП та ВЧ – Шеннона, колізій та мінімальної ентропій. Вважається, що обґрунтовані та запропоновані нижче методики оцінки повинні (можуть) бути застосовані щодо ДШ (генераторів) як фізично справжніх (PTRNG), так і нефізично справжніх (NPTRNG) ДШ.

2.2.3.1 Оцінка ентропії Шеннона

Для дискретної випадкової змінної X з можливими значеннями $x_0, x_1, \dots, x_{k-1}$ та відповідними ймовірностями появи символів $(p_0, p_1, \dots, p_{k-1})$,

ентропія Шеннона визначена як

$$H(\mathrm{X}) = -\sum_{i=0}^{k-1} p_i \cdot \log\, p_i. \tag{2.6}$$

Надалі позначатимемо частоту появи символу x_i у вибірці розміру n як $n(x_i)$ та відповідну емпіричну ймовірність $p_i = n(x_i)/n$.

Якщо застосовувати формулу (2.6) напряму до реальних статистичних даних, то оцінка буде зміщеною [44, 45]. Щоб отримати незміщену оцінку, можливо використовувати корекцію, зокрема наступні методи корекції, що задані формулами (2.7)-(2.9) [46]:

$$\hat{H}(X) = H(X) + \frac{m-1}{2n}, \tag{2.7}$$

де m – кількість різних символів, що зустрілися в статистичних даних, n – розмір вибірки;

$$\hat{H}(X) = n \cdot H(X) - \frac{n-1}{n} H_{-i}(X), \tag{2.8}$$

де $H(\mathrm{X})_{-i}$ – це $H(X)$ без доданка $p_i \cdot \log\, p_i$;

$$\hat{H}(X) = -\sum_{i=0}^{n-1} a_i \cdot h_i \tag{2.9}$$

Також $h_i = \sum_{j=1}^{k-1} \left[\left[p_j \cdot n == i\right]\right]$ та $a_i = -\frac{i}{n}\log\frac{i}{n} + \left(\frac{1-\frac{i}{n}}{2n}\right)$ (тут [[•]] позначає предикат). Для зручності надалі будемо використовувати позначення MM для формули (2.8), JKF для формули (2.8) та BUB для формули (2.9).

Інший підхід до усунення зміщення полягає у використанні формули Баєса [47]:

$$\hat{H}(X) = -\sum_{i=0}^{k-1} \frac{p_i \cdot n + a_i}{n + A} \log\left(\frac{p_i \cdot n + a_i}{n + A}\right), \tag{2.10}$$

де $A = \sum_{i=0}^{k-1} a_i$, а значення a_i обираються в залежності від конкретного метода. Зокрема, часто $a_0 = a_1 = \cdots a_{k-1} = \mathrm{const}$. Найбільш популярні вибори констант наступні [44, 47]:

$a_i = 1/2;$

$a_i = 1;$

$a_i = 1/k;$

$a_i = \sqrt{n}/k.$

3 експериментальних досліджень у роботах [44, 48, 49] також варто виділити наступні три перспективні підходи до оцінки ентропії Шеннона:

Підхід, що був запропонований у роботі [50] (надалі – SHU-оцінка). Значення ентропії визначається за формулою

$$\hat{H}(X) = \psi(n) - \frac{1}{n}\sum_{i=0}^{k-1}\left(\psi(p_i \cdot n) + (-1)^{p_i \cdot n}\int_0^{\frac{1}{\xi}-1}\frac{t^{p_i \cdot n-1}}{1+t}dt\right). \quad (2.11)$$

Підхід, що був запропонований у роботі [51] (надалі – CS-оцінка). Значення ентропії визначається за формулою

$$\hat{H}(X) = -\sum_{i=0}^{k-1}\frac{\tilde{p}_i \log \tilde{p}_i}{1-(1-\tilde{p}_i)^n}, \quad (2.12)$$

де $\tilde{p}_i = \left(1 - \frac{m}{n}\right)p_i$.

Підхід, що був запропонований у роботі [48] (надалі – SHR-оцінка). Значення ентропії визначається за формулою

$$\hat{H}(X) = -\sum_{i=0}^{k-1}\tilde{p}_i \log \tilde{p}_i, \quad (2.13)$$

де $\tilde{p}_i = \lambda/k + (1-\lambda)\mathrm{p}_i$, причому

$$\lambda = \frac{1-\sum_{i=0}^{k-1}(p_i)^2}{(n-1)\sum_{i=0}^{k-1}(1/k - p_i)^2}.$$

Також можливо виділити ряд інших методів оцінки ентропії Шеннона, що використовують більш складні статистичні методи, проте вони не набули широкого поширення через складність реалізації.

2.2.3.2 Оцінка колізійної ентропії ДШ

Колізійна ентропія визначена наступним чином:

$$H_2(X) = -\log\left(\sum_{i=0}^{k-1} p_i^2\right). \quad (2.14)$$

Для колізійної ентропії відомі наступні обмеження нерівностями:

$$H_{\min} \le H_2 \le H, \quad (2.15)$$

$$H_{\min} \le H_2 \le 2H_{\min}. \quad (2.16)$$

Для практичних задач нерівностей, на нашу думку, достатньо, щоб оцінити значення колізійної ентропії, маючи оцінки мінімальної ентропії та ентропії Шеннона.

Для більш точних оцінок можливо скористатися методом, що описаний у роботі [45]. Метод залежить від двох параметрів – параметра точності оцінки δ та параметра статистичної помилки ò. Нехай існують деяка константа M, що гарантовано має значення більше за будь-яке можливе значення колізійної ентропії та довільна константа $c > 0$. Наприклад, $M = H(X)$ або $M = 2H_{\min}(X)$. Тоді, можливо використовувати наступний алгоритм:

1) Обрати розмір блоку $N = \lceil c \cdot 2^{M/2}\delta^{-2} \rceil$
2) Обчислити кількість блоків $l = \lfloor n/N \rfloor$
3) Для $j = 1, \dots, n/N$
   a. Для $i \in \{(j-1)N + 1, \dots, (j-1)N\}$
      i. $n(x_{i+1}) = n(x_i) + 1$
   b. $q_j = \frac{1}{m(m-1)} \left(\sum_{i=0}^{k-1} n(i)^2 - m \right)$
4) Знайти медіану q послідовності $q_1, \dots, q_l$
5) Повернути $-\log q$

Метод доказово оцінює значення колізійної ентропії з похибками (δ,ò), якщо $c \geq \log(1/\text{o})$.

2.2.3.3 Оцінка мінімальної ентропії ДШ (ВП)

Для дискретної випадкової змінної X з можливими значеннями $x_0, \text{x}_1, \dots, x_{k-1}$ (та відповідними ймовірностями $p_0, \text{p}_1, \dots, p_{k-1}$) мінімальна ентропія визначена як

$$H_{\min}(X) = -\log_2 \left(\max_{1 \leq i \leq k} \{\text{p}_i\} \right). \quad (2.17)$$

Окрім позначення $H_{\min}(\cdot)$ для мінімальної ентропії, також є популярним позначенням $H_{\infty}(\cdot)$.

Для оцінки мінімальної ентропії можливо виділити два основних підходи. Перший підхід базується на ентропійній статистиці, вперше описаний в [52]. Другий підхід базується на предикторах (англ. predictor) (на основі прогнозування), вперше описаних у [53].

Ентропійна статистика призначена для обчислення окремої статистики на вибірках. До методів, що використовують ентропійну статистику, належать колізійний тест, тест на стиснення, тест Маркова.

Хоча оцінювачі ентропії (за винятком тесту Маркова) спочатку були розроблені для застосування до незалежних виходів, тести показали хороші результати при застосуванні до даних із залежностями.

Оцінювачі ентропії припускають, що розподіл ймовірностей описує вихід випадкового джерела шуму, але розподіл ймовірностей невідомий. Метою кожного оцінювача є виявлення інформації про невідомий розподіл на основі статистичних вимірювань.

Тести колізій та стиснення розв'язують рівняння для невідомого параметра, де рівняння є різними для кожного оцінювача. Ці рівняння походять із очікуваного значення цільової статистики з використанням майже рівномірного розподілу, який забезпечує нижню межу мінімальної ентропії. Майже рівномірний розподіл є прикладом однопараметричного сімейства розподілів ймовірностей, параметризованих p, P_p:

$$P_p(i) = \begin{cases} p, \text{якщо i} = 0 \\ \frac{1-p}{k-1}, \text{інакше} \end{cases}, \tag{2.18}$$

де k – кількість станів у вихідному просторі, а $p \geq \frac{1-p}{k-1}$, що має місце, коли $p \geq 1/k$.

Іншими словами, один вихідний стан має максимальну ймовірність, а решта вихідних станів рівноймовірні.

Підхід на основі предикторів використовує два показники для отримання оцінки. Перший показник базується на глобальній продуктивності предиктора P_{global}, яка в літературі з машинного навчання називається точністю. По суті, предиктор фіксує частку правильних припущень. Це приблизно вказує на те, наскільки добре можна очікувати, що предиктор вгадає наступний вихід із джерела шуму на основі результатів довгої послідовності припущень. Другий показник P_{local} базується на найбільшій кількості правильних передбачень у рядку, який називається локальним показником ефективності. Ця метрика

корисна для виявлення випадків, коли джерело шуму переходить у дуже передбачуваний стан протягом деякого часу, але предиктор може не працювати добре на довгих послідовностях. Розрахунки для оцінки локальної ентропії походять з теорії ймовірностей пробігів і повторюваних подій [54].

Для того, щоб оцінки предиктора схилялися до консервативної недооцінки мінімальної ентропії, P_{global} замінюється на P'_{global}, що відповідає 99-му квантилю кількості правильних прогнозів на основі спостережуваної кількості правильних прогнозів. Зауважимо, що порядок, у якому відбуваються правильні прогнози, не впливає на оцінку мінімальної ентропії на основі P_{global}. Наприклад, прогноз завжди може бути правильним для першої половини вихідних даних у наборі даних і завжди неправильним для другої половини вихідних даних. Оцінка мінімальної ентропії цієї послідовності на основі P_{global} становить половину довжини даних у бітах.

3 іншого боку, для іншої послідовності предиктор може мати 50% шанс бути правильним для кожного зразка в цій послідовності. Мінімальна оцінка ентропії цієї другої послідовності, заснована на P_{global}, така ж, як і для першої послідовності. Однак типова тривалість успішного прогнозування для цих двох послідовностей дуже різна. Таким чином, цей підхід враховує ефективність локального передбачення, щоб консервативно зменшити оцінку мінімальної ентропії, якщо спостережувана поведінка локального передбачення є статистично значущою, враховуючи глобальний рівень успіху передбачення. Оцінки предикторів досягають цього, базуючи оцінку мінімальної ентропії на $\max(P'_{\text{global}}, P_{\text{local}})$, де P_{local} це частка успішного прогнозу, для якої спостережуваний найдовший ряд правильних прогнозів становить 99%. Фактично це одностороння перевірка гіпотези, яка відхиляє P'_{global} на користь P_{local}, якщо спостережуваний найдовший пробіг, враховуючи ймовірність успіху P'_{global}, перевищує 99%.

2.2.3.4 Загальні рекомендації до оцінки ентропії Шеннона, колізійної та мінімальної ентропії

Далі зведено дані експериментальних досліджень ентропії (рис. 2.2).

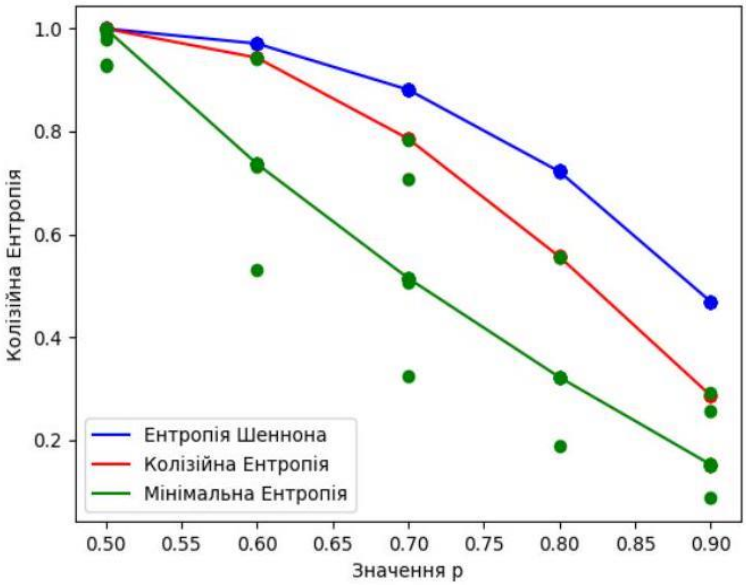

Рисунок 2.2 – Експериментальна оцінка ентропії

З отриманих даних випливає, що ентропію Шеннона та колізійну ентропію можливо оцінити доволі точно, у той час як мінімальну ентропію оцінити складніше. Проте, мінімальна ентропія є важливим показником для багатьох практичних застосувань, тому має сенс доповнювати оцінки мінімальної ентропії оцінками ентропії Шеннона та колізійної ентропії, якщо спостерігається розбіжність в отриманих оцінках мінімальної ентропії. Це є актуальним здебільшого для низькоентропійних джерел шуму.

2.3 Висновки до розділу

У цьому розділі було розглянуто методи та засоби аналізу, оцінки та порівняння властивостей ВП. Основна увага приділена статистичним і стохастичним методам тестування ГВЧ.

Статистичні методи аналізу, такі як тести DIEHARD, NIST STS та DIEHARDER, дозволяють оцінювати ГВЧ на відповідність криптографічним вимогам. Критерії прийняття рішень щодо випадковості базуються на порівнянні статистичних характеристик послідовностей з теоретичними очікуваннями.

Використання рівнів значущості дозволяє контролювати ймовірність хибних висновків, зокрема помилок 1-го та 2-го роду.

Методика тестування NIST STS передбачає застосування 16 статистичних тестів для виявлення відхилень від випадкового розподілу. Визначення випадковості ґрунтується на оцінці ймовірностей (p-значень) та побудові статистичних портретів генераторів.

Набір тестів DIEHARD містить 15 різних статистичних перевірок, що аналізують різні аспекти випадковості. Його подальша модифікація у вигляді DIEHARDER розширила можливості тестування та підвищила його точність.

Стохастичні методи аналізу випадковості доповнюють статистичні, оскільки вони дозволяють оцінювати початкову ентропію джерел випадковості. Доведено, що відповідність стандартам NIST SP 800-90B є важливим фактором при виборі надійних генераторів випадкових чисел.

Методи оцінки ентропії базуються на використанні ентропії Реньї та її похідних. Вони дозволяють визначити рівень непередбачуваності вихідних даних генератора і підтвердити його відповідність вимогам криптографічної стійкості.

Загалом, результати аналізу показують, що комплексне використання статистичних і стохастичних методів є необхідною умовою для оцінки якості генераторів випадкових послідовностей, що застосовуються у криптографічних системах.

3 МЕТОДИ ТА АЛГОРИТМИ ГЕНЕРУВАННЯ ТА АНАЛІЗУ ВП ТА ПВП НА ОСНОВІ ДНК

У якості джерела випадковості можна використовувати певні фізичні чи нефізичні явища. Цікавим напрямком, що стосується нових ДШ, є використання ДНК. Ці макромолекули, фактично являючи собою унікальну для кожної живої істоти послідовність, в теорії, через це можуть слугувати джерелом випадковості, яке в подальшому можна використовувати в процесі генерації ВП та ПВП. Список наявних у відкритому доступі досліджень по цій тематиці мізерно малий, тому перспективність та новизна розробки такого підходу – на дуже високому рівні.

3.1 Подання ДНК-послідовності у вигляді бінарних даних

Початковими даними для вирішення даної задачі є ДНК, яке пропонується використовувати у якості нефізично справжнього ДШ. ДНК складається з чотирьох типів азотистих основ: аденіну (A), цитозину (C), гуаніну (G) та тиміну (T). Отже, ДНК-послідовність можна розглядати як ряд символів над алфавітом з чотирма можливими значеннями (рис. 3.1).

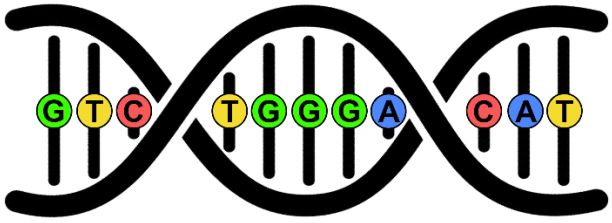

Рисунок 3.1 – Дезоксирибонуклеїнова кислота у вигляді азотистих основ

Для того, щоб використати ДНК як джерело двійкової випадковості, необхідно відобразити цю послідовність у бінарний вигляд [55]. Найпростіший спосіб – закодувати кожну основу парою бітів. Оскільки потужність алфавіту ДНК дорівнює 4, а двійкового – 2, то для унікального взаємно-однозначного відображення достатньо двобітної комбінації, оскільки $2^2=4$.

Тоді бінарне представлення формується наступним чином: азотистим основам A, C, G, T ставляться у відповідність двобітні комбінації як наведено у

табл. 3.1. відповідно. Саме таке призначення кодів зумовлено алфавітним впорядкуванням основ (A, C, G, T) та, відповідно, зростаючий числовий порядок бінарних кодів. Отже, чим раніше у алфавіті зустрічається комбінація, тим меншому значенню вона відповідатиме, що забезпечує однозначність і зручність кодування.

Таблиця 3.1 – Двійкові комбінації для представлення алфавіту ДНК

Азотиста основа з ДНК	Двійкова комбінація для її представлення
A	00
C	01
G	10
T	11

Згідно з цією таблицею, будь-яка ДНК-послідовність може бути переведена у бітовий потік. При цьому довжина двійкової послідовності буде вдвічі більшою за кількість основ вихідної ДНК. Наприклад, послідовність ДНК фрагменту *Homo sapiens* “GTCTGGGACAT” (рис. 3.1) після застосування правил табл. 3.1 відобразиться бітовим рядком:

$$\text{GTCTGGGACAT} \longrightarrow 10110111101010000100112$$

Таким чином, ми отримуємо сиру двійкову послідовність з ДНК довільної довжини. Зразки реальних геномних послідовностей різних організмів для експериментів були отримані із відкритих баз даних (наприклад, GenBank NCBI [56] та DDBJ [57] у вигляді текстових файлів), після чого переведені у .dat-файли з бітами за наведеним правилом. Отримані “сирі” бітові дані на основі ДНК шаблонів далі слугуватимуть вхідним матеріалом для генератора.

У роботі представлені результати досліджень тільки для ДНК послідовностей людини, оскільки отримані для інших організмів результати є схожими, що додатково свідчить про однакову природу походження ДНК, а отже про можливість розгляду ДНК у якості нефізичного джерела шуму. Також, як уже згадувалося раніше, ДНК є унікальною і отримані на її основі ПВП чи ВП можуть бути використані як унікальні дані. Отже отримавши зразок власного ДНК, людина зможе використовувати його для отримання, наприклад, унікальних секретних ключів, тощо.

3.2 Методика аналізу та тестування послідовностей

Для використання послідовностей необхідно мати можливість оцінити їх властивості. Вихідна бінарна послідовність, без додаткового опрацювання, може не відповідати критеріям випадковості. Теоретичні засади статистичного та стохастичного дослідження послідовностей наведені попередньо у роботі (див. розділ 2). Пропонується алгоритм оцінювання властивостей, який є релевантним для будь-яких двійкових даних. Виконання алгоритму передбачає проведення стохастичного та статистичного тестування послідовностей. Принцип роботи алгоритму зображено на блок-схемі (рис. 3.2).

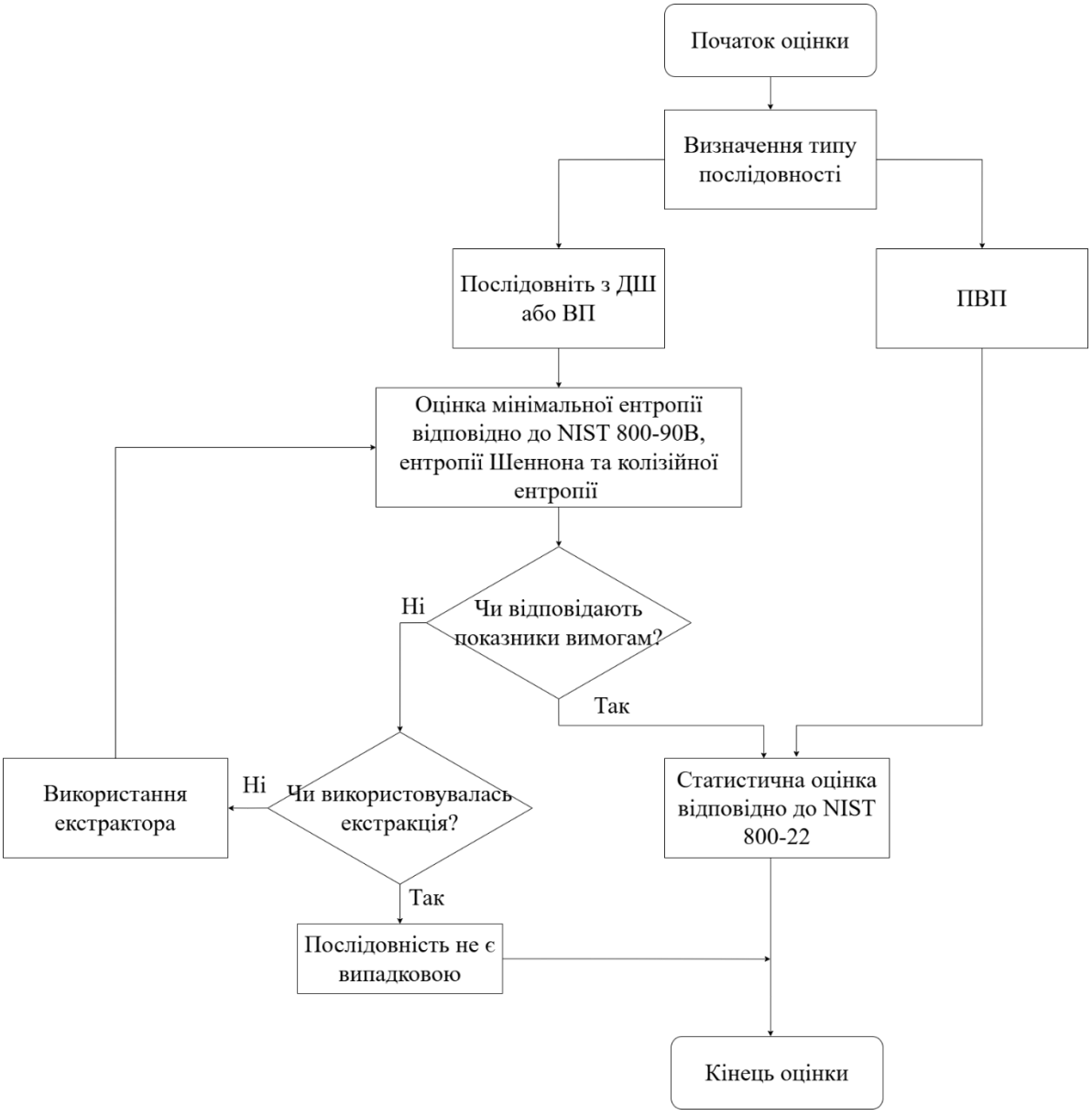

Рисунок 3.2 – Блок-схема алгоритму оцінювання властивостей послідовностей

Отже, запропонований алгоритм оцінювання дає змогу залежно від типу досліджуваної послідовності визначати значення її ентропійних і статистичних характеристик. На основі отриманих оцінок можна ухвалювати рішення щодо подальшої роботи з послідовністю: прийняття, удосконалення або відхилення.

3.3 Дослідження «сирих» послідовностей

Результати стохастичного та статистичного тестування «сирої» послідовності з ДНК Homo Sapiens (людини) надаються у табл. 3.2, та на рис. 3.3 відповідно. Для перевірки показників ентропії використовується фрагмент послідовності довжиною 1 МБ (у двійковому представленні), а для статистичного тестування – фрагмент довжини 13 МБ (у двійковому представленні).

Таблиця 3.2 – Результати тестування послідовності з ДНК Homo Sapiens методами ентропій

Оцінка мінімальної ентропії за [38]		Оцінка ентропії Шеннона методами з [44]	
MCVBitTest	0.992654	MaxLikehood	0.968398
CollisionTest	0.791030	MM	0.968413
MarkovTest	0.884869	UnveiIJ	0.968422
CompressionTest	0.328646	BUB	0.968413
MMCinWindowBitTest	0.273780	Bayes a=1,0	0.968413
LagPredictionBitTest	0.038148	Bayes a=1/k	0.968398
MultiMCPredictionBitTest	0.179322	Bayes a = sqrt(n)/k	0.968460
LZ78YPredictionBitTest	0.202291	CS	0.968205
TupleTest	0.032402	SHR	0.968434
LRSTest	0.000022	SHU	0.968420
Оцінка колізійної ентропії за [45]			
Naive		0.936559	
Unbiased block = 1		0.936590	
Unbiased block = 10		0.870314	

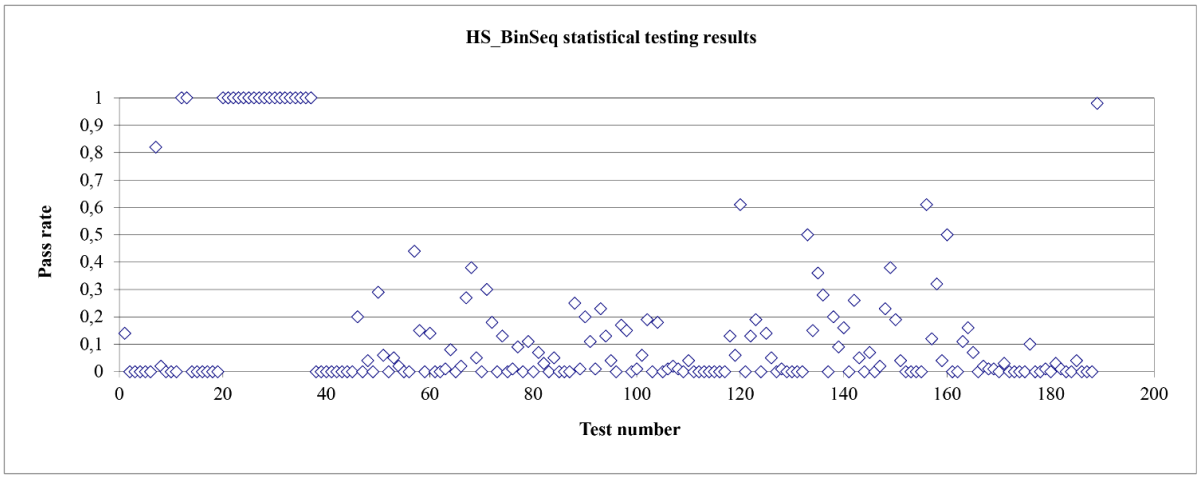

Рисунок 3.3 – Статистичний портрет послідовності з ДНК Homo Sapiens

Дійсно, статистичний аналіз показує, що такі послідовності мають значні відхилення від рівномірності розподілу та інші дефекти, успадковані від структури геному. Практично майже всі тести стандартного статистичного набору NIST STS провалюються для сирих ДНК-послідовностей. Стохастична (ентропійна) оцінка також виявляє недостатню ентропію: присутні передбачувані шаблони, повторюваність і нерівномірність, що підтверджується низькими значеннями показників мінімальної та колізійної ентропії (див. розд. 2) для сирих даних. Отже, сиру двійкову послідовність з ДНК необхідно покращити перед використанням у криптографічних перетвореннях. Для вилучення “якісної” випадковості з такої послідовності пропонується прибігнути до алгоритму екстракції.

3.3 Генерування випадкових та псевдовипадкових послідовностей з ДНК

Для покращення властивостей двійкових даних, отриманих з ДНК, використано метод криптографічної екстракції на базі блочного шифру. Ідея полягає в тому, щоб “перемішати” або розподілити біти сирої послідовності рівномірно, використовуючи стійке криптографічне перетворення. Відомо, що застосування перевірених криптографічних примітивів (геш-функцій, шифрів) як функцій кондиціювання рекомендується стандартами NIST SP 800-90A/90B [58, 38]. Зокрема, у NIST SP800-90B наведено конструкцію ентропійного екстрактора

з використанням блочного або потокового шифру, що гарантує покращення стохастичних властивостей вихідної вибірки.

В рамках дослідження в якості екстрактора випадковості обрано український стандарт блочного симетричного шифру ДСТУ 7624:2014 (Калина) у режимі лічильника (CTR). Калину в режимі CTR використано як генератор псевдовипадкових бітових послідовностей (DRBG), де початковим станом слугують дані з ДНК. Такий вибір зумовлено тим, що алгоритм ДСТУ 7624:2014 відповідає сучасним вимогам криптостійкості, а також підтримує різні розміри блоків та ключів (128 або 256 біт), що надає гнучкість. Крім того, CTR-режим блочного шифру забезпечує потокове генерування великого обсягу випадкових даних із відносно невеликого початкового ключа (seed), що відповідає задачі побудови ГВП. Для обчислення ПВП стандартом пропонується задання ключа та nonce як гешу від внутрішнього стану генератора, а для обчислення ВП – використання випадкових рядків відповідної довжини. Такі рядки можуть бути отримані з використанням, наприклад, довірених PTRNG (Quantis QRNG (Switzerland) [59], Гряда-3 (Україна) [60]) або NPTRNG. Схема процесу отримання послідовностей з використанням екстрактора наведена на рис. 3.4.

Рисунок 3.4 – Процес генерації ПВП та ВП

Алгоритм ComputePRS, послідовність дій якого наведена у додатку Г, дозволяє генерувати ПВП заданої довжини на основі бінарних ДНК-фрагментів

(або інших бінарних даних форматів .dat, .bin тощо) та заданих параметрів ключа і nonce. Початкові значення ключів, nonce і ДНК-фрагментів наведені в табл. 3.3.

Таблиця 3.3 – Значення вхідних даних алгоритму обчислення ПВП

Ключ (256 біт)	Nonce (256 біт)	Фрагмент ДНК у двійковому вигляді, взятий за основу (довжини 1МБ)
0xFF, 0x06, 0x21, 0xA6, 0xA0, 0x28, …, 0x02, 0x4D	0x73, 0xA4, 0x12, 0xEA, 0xD0, 0x45, …, 0xFE, 0x3F	0xD5, 0x76, 0x48, 0xF7, 0x10, 0x02, 0x2E, 0xFD, …
		0x80, 0xFA, 0x43, 0x10, 0xF2, 0x2F, 0xD1, 0x32, …
		0x51, 0x4E, 0x7C, 0x9C, 0x3F, 0xFB, 0x3F, 0xF2,…

Алгоритм генерації ВП ComputeRS, послідовність дій якого також наведена у додатку Г, відповідає вимогам ДСТУ 7624:2014 щодо генерації. Ключ і nonce для кожної ітерації шифрування отримуються за допомогою генератора /dev/random [61], реалізованого на ядрі операційної системи. Цей генератор належить до класу NTG.1, тобто є нефізично справжнім генератором. Фрагменти ДНК, що використовуються для отримання ВП, аналогічні тим, які застосовуються для ПВП (табл. 3.3).

Його реалізація забезпечує отримання ВП завдяки використанню нефізично справжнього генератора, але має меншу швидкодію через необхідність щоразу генерувати новий ключ і nonce із зовнішнього джерела (/dev/random). Проте це покращує безпеку, ускладнюючи розрахунок попередніх та наступних значень ключів і nonce.

3.4 Аналіз випадкових послідовностей ентропійними методами

Вхідними даними є ВП, отримані екстракцією. Завдання полягає у застосуванні ентропійних тестів до цих ВП і прийнятті рішень щодо їх якості випадковості.

Тестування виконувалося для даних у форматі двійкових файлів (.dat) з послідовностями довжини 13 МБ (13 631 488 байтів або приблизно 10^8 біт) та 120 МБ (125 829 120 байтів або приблизно 10^9 біт).

Результати ентропійних перевірок наведені у таблиці 3.4.

Таблиця 3.4 – Результати тестування ВП ентропійними методами

Послідовність	Мінімальна ентропія	Ентропія Шеннона	Колізійна ентропія
13MB_RandSeq1	0.9959725	0.999983	0.999996
13MB_RandSeq2	0.9959991	0.998023	0.999997
13MB_RandSeq3	0.9962625	0.998302	0.999996
120MB_RandSeq1	0.9983775	0.994104	0.999998
120MB_RandSeq2	0.9986189	0.999989	0.999998
120MB_RandSeq3	0.9986769	0.999989	0.999998

Результати вказують на те, що ентропійні показники отриманих ВП повністю відповідають необхідним вимогам непередбачуваності. Оскільки отримані послідовності успішно пройшли стохастичне тестування, наступним етапом алгоритму оцінювання є аналіз їхніх статистичних властивостей за допомогою набору тестів NIST. Статистичне тестування виконувалося як для ВП, так і для ПВП.

3.5 Аналіз статистичних властивостей отриманих послідовностей

Вхідними даними є ВП та ПВП, отримані екстракцією. Рішення поставленої задачі полягає у застосуванні статистичних тестів набору NIST STS [33] до отриманих послідовностей.

Результати статистичного аналізу наведені в табл. 3.5, 3.6. У таблицях подано такі дані:

- довжина та ідентифікатор послідовності;
- кількість тестів з набору, де успішність проходження становила понад 96% бітових потоків (>96%);
- кількість тестів, де успішність проходження становила понад 99% бітових потоків (>99%);
- кількість тестів, для яких значення $p<0.001$;
- загальний статус проходження тестування (Status).

Тестування здійснювалось для послідовностей, збережених у двійковому форматі (.dat), з розмірами 13 МБ (13 631 488 байтів або близько 108 біт) та 120

МБ (125 829 120 байтів або близько 109 біт. Для кожної послідовності було використано 100 бітових потоків.

Таблиця 3.5 – Результати статистичного аналізу ПВП

Послідовність	>96 %	>99 %	p<0.001	Status
13MB_PseudoRandSeq1	188 (99 %)	129 (68 %)	0	Success
13MB_PseudoRandSeq2	189 (100 %)	127 (67 %)	0	Success
13MB_PseudoRandSeq3	189 (100 %)	137 (72 %)	0	Success
120MB_PseudoRandSeq1	189 (100 %)	112 (59 %)	0	Success
120MB_PseudoRandSeq2	189 (100 %)	95 (50 %)	1	Success
120MB_PseudoRandSeq3	189 (100 %)	98 (52 %)	1	Success

Таблиця 3.6 – Результати статистичного аналізу ВП

Послідовність	>96 %	>99 %	p<0.001	Status
13MB_RandSeq1	189 (100 %)	132 (70 %)	0	Success
13MB_RandSeq2	188 (99 %)	118 (62 %)	0	Success
13MB_RandSeq3	188 (99 %)	124 (66 %)	0	Success
120MB_RandSeq1	189 (100 %)	94 (50 %)	1	Success
120MB_RandSeq2	189 (100 %)	113 (60 %)	0	Success
120MB_RandSeq3	189 (100 %)	104 (55 %)	1	Success

Як видно з таблиць, результати для деяких послідовностей за критерієм 96% (13MB_PseudoRandSeq1, 13MB_RandSeq2 і 13MB_RandSeq3) вказують на неповне проходження набору тестів. Це пояснюється тим, що доля проходження одного з тестів набору для цих послідовностей трохи нижча за межу, рекомендовану NIST для такої кількості бітових потоків.

NIST пропонує кілька підходів для інтерпретації отриманих результатів. Серед них є аналіз долі послідовностей, що проходять тест, і аналіз розподілу P-значень на однорідність. Якщо ці методи не дають чіткого висновку, рекомендується провести додаткові дослідження на інших вибірках генератора для визначення того, чи є отриманий результат статистичною аномалією або ознакою невипадковості.

Отже, неуспішність одного тесту деякими послідовностями може розглядатися як статистична похибка, оскільки результати інших послідовностей, зокрема більших розмірів (які дають точнішу оцінку),

демонструють повну успішність проходження тестів. Крім того, у [33] зазначено, що межа проходження є приблизною.

Окрему увагу слід приділити статистиці P-значень для тестів, які було провалено. Наприклад, для тесту «Перевірка шаблонів, що не перекриваються» послідовності 13MB_PseudoRandSeq1 значення проходження становить 0.9541, а P-значення – 0.7887, що значно перевищує рівень значущості 0.001. Аналогічно для послідовностей 13MB_RandSeq2 і 13MB_RandSeq3 значення долі проходження також 0.9541, а P-значення відповідно 0.2826 та 0.0830. Це підтверджує наявність статистичної аномалії.

Крім того, кількість результатів тесту «Перевірка шаблонів, що не перекриваються» в загальній статистиці становить 148 значень. Відповідно, незначні відхилення окремих результатів можна розглядати як середні значення в межах загальної кількості тестів, що відповідає вимогам. Варто зазначити, що для послідовностей довжиною 10^9 біт такі статистичні аномалії для рівня 96% не спостерігаються, що свідчить про недостатність довжини 10^8 біт для точної статистичної оцінки.

Хоча статистика проходження тестів (>96%) для різних довжин не має суттєвих відмінностей, за критерієм 99% більші послідовності мають більш щільний і рівномірний розподіл, що підтверджують статистичні портрети, представлені у додатку Д.

Статистичний аналіз свідчить, що отримані за допомогою екстрактора ПВП і ВП відповідають вимогам статистичного тестування. Таким чином, ПВП успішно проходять статистичні тести, а ВП успішно проходять як стохастичні, так і статистичні випробування.

3.6 Висновки до розділу

Аналіз вибірки ДНК з використанням алгоритму засвідчив невідповідність її статистичних характеристик необхідним вимогам. Це обумовлено значним відхиленням ДНК-послідовностей від рівномірності через їх специфічну структуру та властивості. ДНК містить інформацію щодо структури РНК та білків, які зазвичай повторюються або формують групи, що спричиняє

виникнення шаблонів у послідовностях. Результати стохастичних випробувань свідчать про наявність певного рівня ентропії у послідовностях, що дозволяє застосовувати їх для екстракції випадковості та подальшого отримання ПВП/ВП.

Для вирішення недоліків існуючих досліджень пропонується алгоритм, що дає змогу отримувати ПВП та ВП із використанням невеликого об'єму вхідних ДНК-даних. Алгоритм реалізований відповідно до стандарту блокового симетричного шифрування ДСТУ 7624:2014 у режимі CTR. Подібний криптографічний примітив рекомендований у якості компонента посилення у стандарті NIST 800-90A. Процес генерації та результати визначаються особливостями криптографічної конструкції перетворення, що забезпечує отримання довгих ПВП та ВП із коротких вхідних даних.

Описаний процес формування послідовностей на базі ДНК: на відміну від підходів [62-64], де отримуються «сирі» ДНК-послідовності, які не відповідають сучасним вимогам, розроблені алгоритми шляхом екстракції випадковості дозволяють формувати псевдовипадкові та випадкові послідовності, що задовольняють необхідним вимогам. Перевагою нових алгоритмів у порівнянні з існуючими є можливість отримання випадкових даних майже необмеженої довжини на основі коротких вхідних даних, чого неможливо досягти в [62-64]. Додатковою особливістю формування ВП, що підвищує безпеку алгоритму та одержуваних даних, є використання випадкових сеансових ключів і значень nonce, які отримуються за допомогою NPTRNG /dev/random. Це забезпечує надійний захист від компрометації завдяки тому, що на кожній ітерації ці значення генеруються абсолютно випадково та незалежно, не містячи жодної інформації про попередні значення.

Результати стохастичних тестів ВП, а також статистичних тестів ПВП і ВП демонструють, що отримані послідовності, на відміну від результатів робіт [62-64], є рівномірними, позбавленими статистичних недоліків і характеризуються високим ступенем випадковості. Це підтверджує можливість використання таких послідовностей у криптографічних перетвореннях та процедурах.

4 ПОРІВНЯННЯ ТА ПОДІБНІСТЬ ВП ТА ПВП НА ОСНОВІ ДНК

Важливим питанням є також взаємна подібність послідовностей, згенерованих із різних зразків ДНК або з одного зразка при використанні різних ключів та параметрів, оскільки необхідно забезпечити їхню унікальність та відсутність взаємної кореляції. Початковими даними для вирішення цієї задачі є існуючі алгоритми оцінки подібності послідовностей, які мають певні недоліки з погляду ефективності. Для їх усунення передбачено застосування оптимізованих структур і типів даних, що забезпечують економію ресурсів пам’яті, а також універсальних криптопримітивів, здатних працювати у різних режимах. Як результат вирішення задачі передбачається отримати більш ефективні алгоритми, що дозволяють заощаджувати ресурси та/або мають розширені функціональні можливості.

4.1 Оцінка подібності послідовностей методом k-мерів

Одним із простих та ефективних способів оцінки схожості послідовностей без вирівнювання є підхід, що базується на підрахунку k-мерів. Хоча основне застосування цього методу традиційно відбувалося в області ДНК, у даній роботі розглядається його використання для порівняння двійкових послідовностей (ПВП та ВП). У біоінформатиці k-мери представляють собою підрядки довжини k певної біологічної послідовності [65]. Аналогічно, для бітових, байтових чи інших алфавітів k-мери визначаються як послідовності з відповідних елементів.

Ілюстрацію підрахунку k-мерів у ДНК наведено у табл. 4.1, де для короткого фрагмента послідовності «GTCAGC» наведено всі можливі k-мери для значень k від 1 до 6.

Таблиця 4.1 – Значення можливих k-мерів для фрагменту ДНК

Послідовність: GTCAGC	
Довжина *k*-мерів	*k*-мери
1	G, T, C, A, G, C
2	GT, TC, CA, AG, GC
3	GTC, TCA, CAG, AGC
4	GTCA, TCAG, CAGC
5	GTCAG, TCAGC
6	GTCAGC

Для додаткової наочності на рис. 4.1 представлено графічне розбиття послідовності на k-мери довжиною 3.

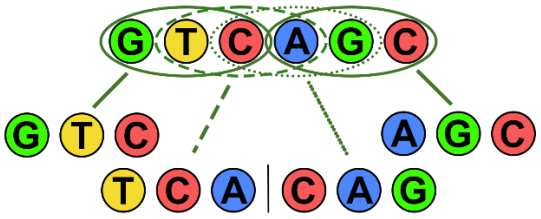

Рисунок 4.1 – Розбиття фрагменту ДНК на *k*-мери довжини 3

Розбиття послідовності на k-мери дозволяє аналізувати окремі фрагменти фіксованого розміру, замість аналізу всієї послідовності цілком. Ефективність цього методу обґрунтовується простотою операцій над множинами та широкою доступністю алгоритмів для їх обробки.

Для правильного вибору довжини k-мерів слід орієнтуватися на розмір досліджуваної послідовності та характер алфавіту. У [66] пропонується розраховувати необхідну довжину за наступною формулою:

$$p = \frac{1}{\frac{(\bar{\Sigma})^k}{g}+1} < 0.01 \tag{4.1}$$

де g – розмір геному або послідовності, а Σ – алфавіт. Проте ця формула більш придатна для двійкового алфавіту або при використанні алгоритму MinHash. При порівнянні ДНК з використанням k-мер відстані можливим є вибір довжини k таким чином, щоб розмір простору всіх можливих k-мерів перевищував розмір досліджуваного геному:

$$(\bar{\Sigma})^k > g \tag{4.2}$$

На основі цього пропонується алгоритм KmerDistCount оцінки подібності двох послідовностей, реалізація якого наведена у додатку Е.

Першим удосконаленням порівняно з алгоритмом, описаним у [67], є конвертація k-мерів у 64-бітні числа (Крок 2). Це значно спрощує порівняння, оскільки числові операції виконуються швидше, ніж посимвольне порівняння рядків. Крім того, збереження 64-бітних чисел вимагає набагато меншого обсягу пам’яті, ніж зберігання рядкових представлень, що особливо помітно при

великих значеннях k. Наприклад, один k-мер довжиною 32, представлений як ціле число, займає 8 байт, у той час як рядкове представлення – 32 байти. Варто зазначити, що конвертація до 64-бітних чисел доцільна для k<33 при аналізі ДНК та для k<65 при роботі з двійковими послідовностями; при перевищенні цих меж використовуються звичайні рядкові представлення, як це робиться у [67], щоб уникнути проблеми переповнення 64-бітних чисел.

Другим удосконаленням є формування структури «ключ-значення» з використанням об'єднання лише тих унікальних k-мерів, які присутні хоча б в одній з послідовностей (Кроки 3 та 4). На відміну від підходу у [67], де зберігається весь простір можливих k-мерів, запропонований метод дозволяє уникнути збереження невикористовуваних елементів, що сприяє значному зниженню споживання пам'яті. Наприклад, при порівнянні ДНК послідовностей довжиною 10000 елементів із k=31 максимальна кількість унікальних k-мерів становить 9970 (число елементів послідовності мінус розмір k плюс 1), тоді як алгоритм з [67] створює структуру з повним простором розмірності 4^k, що є недосяжним для сучасних комп'ютерів.

Таким чином, запропоновані оптимізації дозволяють ефективно вирішити основну проблему існуючого алгоритму.

Обчислення k-мер відстані (Крок 7) здійснюється за допомогою формули, запропонованої у [68] і використаної також у [67]:

$$dist(s_1, s_2) = \frac{log(0.1+F)-log(1.1)}{log(0.1)} \tag{4.3}$$

$$F = \sum_{Distinct.length} \frac{min(p(s_1),p(s_2))}{min(len(s_1),len(s_2))-k+1} \tag{4.4}$$

де F – сумарна кількість спільних k-мерів, $p(s)$ – кількість входжень конкретного k-меру у відповідній послідовності, а $len(s)$ – довжина порівнюваної послідовності.

4.2 Оцінка подібності послідовностей методом MinHash

У дослідженні також розглядається алгоритм порівняння послідовностей із застосуванням MinHash. Основною модифікацією, що усуває важливий теоретичний недолік щодо обмеження розміру k-мерів, є використання геш-

функції з розширеним вихідним розміром. Для цього запропоновано застосування національного стандарту гешування ДСТУ 7564:2014. На відміну від алгоритму у [66], який використовує геші розміром 32/64 біти для k-мерів довжиною до 32/64 для двійкового алфавіту та 16/32 для ДНК, запропонований підхід дозволяє використовувати набагато більші k-мери (до 512 для двійкового алфавіту та до 256 для ДНК), що суттєво знижує ймовірність колізій.

Пропонується алгоритм MinHashDistCount, реалізація якого також наведена у додатку Е. Його основна ідея заключається в тому, що чим вища схожість послідовностей, тим більше спільних мінімальних гешів вони матимуть.

Для гешування на Кроці 3 використовується геш-функція «Купина». Оскільки для представлення 64-бітних чисел достатньо 8-байтових значень, 32-байтовий вихід функції «Купина» обрізають до 8 байтів. Завдяки криптографічним властивостям функції «Купина» обрізання не впливає на її характеристики [69, 70]. Хоча для коротких послідовностей використання збільшених гешів із подальшим обрізанням може нести додаткові витрати продуктивності, цей підхід забезпечує універсальність алгоритму.

Схожість наборів гешів вимірюється за допомогою оцінки Жаккара. Індекс Жаккара для наборів A та B обчислюється за наступною формулою [66]:

$$jaccard(A_s, B_s) = \frac{|A_s \cap B_s|}{s}, \tag{4.5}$$

де As та Bs – відповідні набори гешів, причому розмір перетину |As ∪ Bs| дорівнює встановленому розміру набору гешів – s. Обраний розмір впливає на точність представлення послідовності, а також визначає обсяги використовуваних ресурсів і час виконання порівняння. Похибка оцінки MinHash відстані визначається згідно з [66].

Сама MinHash відстань обчислюється наступним чином [66]:

$$mHashDist = \frac{\frac{-log(2.0 \times jaccard)}{(1.0 + jaccard)}}{k}, \tag{4.6}$$

де k – довжина k-мерів.

Запропонований алгоритм дозволяє швидко оцінити схожість послідовностей з точністю, яка залежить від розміру набору гешів, при цьому споживаючи значно менше ресурсів (зокрема, пам’яті), ніж методи, що вимагають попереднього вирівнювання. Крім того, метод MinHash є швидшим за класичну оцінку k-мер відстані завдяки використанню спрощеного представлення послідовності у вигляді компактного набору мінімальних гешів.

4.3 Експериментальна оцінка точності алгоритмів

Оцінка точності запропонованих алгоритмів проводилася як для ДНК послідовностей (використовуючи їх як вибірки), так і для двійкових послідовностей, що генерувалися.

4.3.1 Порівняння ДНК послідовностей методом k-мерів

Для демонстрації роботи алгоритму KmerDistCount були використані ДНК послідовності однакових генів людини та шимпанзе. При порівнянні аналізувалися гени INSR (рецептор інсуліну) та VDR (рецептор вітаміну D). Довжина ДНК послідовностей для гена VDR становила приблизно 63000 елементів, а для INSR – від 181000 до 183000 елементів. Згідно з формулою (4.2) для таких розмірів послідовностей рекомендовано використовувати k-мери довжиною 9. Для практичної оцінки були вибрані різні значення k: k=9 (відповідно до розрахунку), k=5 (за замовчуванням у [67]), а також k=7 і k=11 для детальнішого аналізу залежності точності від розміру k. Схожість визначалася як 1 мінус обчислена k-мер відстань.

Результати обчислення k-мер відстані для генів VDR та INSR подано у табл. 4.2.

Таблиця 4.2 – обчислення k-мер відстані для ДНК генів VDR та INSR людини та шимпанзе

Ген VDR					
П1	П2	k-мер відстань			
		k=5	k=7	k=9	k=11
ДНК людини 1	ДНК людини 2	0.0010403	0.0017395	0.0022834	0.0027967
ДНК людини	ДНК шимпанзе	0.0053960	0.0215622	0.0363721	0.0462452

Продовження таблиці 4.2

Ген INSR					
П1	П2	*k*-мер відстань			
		k=5	*k*=7	*k*=9	*k*=11
ДНК людини 1	ДНК людини 2	0.0012025	0.0026962	0.0039040	0.0049289
ДНК людини	ДНК шимпанзе	0.0038318	0.0181211	0.0413278	0.0573898

Отримані значення свідчать про те, що алгоритм забезпечує коректну оцінку схожості ДНК, причому результати майже відповідають даним, отриманим за допомогою різних методів з попереднім вирівнюванням. Найвищу точність демонструє алгоритм при виборі *k*, який оптимально відповідає довжині послідовності згідно з формулою (4.2).

4.3.2 Порівняння ВП методом k-мерів

Для оцінки схожості ВП було проаналізовано по дві послідовності різної довжини: 2 МБ, 6 МБ та 13 МБ. Вибір розміру послідовностей обумовлено необхідною довжиною k-мерів для їх оцінки – відповідно 31, 33 та 35. Результати обчислення k-мер відстані для цих послідовностей наведено у табл. 4.3. Додатково досліджувалися значення *k*=29 та *k*=37 для перевірки зміни точності.

Таблиця 4.3 – Обчислення *k*-мер відстані для випадкових послідовностей

П1	П2	k-мер відстань				
		k=29	*k*=31	*k*=33	*k*=35	*k*=37
2MB_RandSeq1	2MB_RandSeq2	0.885233	0.967787	0.991931	0.997763	0.999482
6MB_RandSeq1	6MB_RandSeq2	0.731474	0.910465	0.975394	0.993671	0.998408
13MB_RandSeq1	13MB_RandSeq2	0.571947	0.828794	0.948960	0.986212	0.996457

Аналіз отриманих результатів показує, що при оптимально підібраній довжині k-мерів (згідно з відповідною формулою) схожість двійкових послідовностей оцінюється на рівні 1.4–3.3 % (k-мер відстань ≈ 0.97–0.986), що свідчить про майже повну унікальність послідовностей, отриманих за допомогою розроблених алгоритмів.

4.3.3 Порівняння ДНК послідовностей методом MinHash

Результати обчислення MinHash відстані для ДНК генів VDR та INSR людини та шимпанзе за алгоритмом MinHashDistCount наведено у табл. 4.4.

Таблиця 4.4 – Обчислення MinHash відстані для ДНК генів VDR та INSR людини та шимпанзе

Ген VDR						
П1	П2	MinHash відстань				
		k=7	*k*=9	*k*=11	*k*=13	*k*=31
ДНК людини 1	ДНК людини 2	0.00071968	0.00044712	0.00055041	0.000780952	0.000630904
ДНК людини	ДНК шимпанзе	0.00511872	0.00989287	0.0119934	0.0116824	0.0114459
Ген INSR						
П1	П2	MinHash відстань				
		k=7	*k*=9	*k*=11	*k*=13	*k*=31
ДНК людини 1	ДНК людини 2	0.000143072	0.00101369	0.00101679	0.00134195	0.00133643
ДНК людини	ДНК шимпанзе	0.00309769	0.0120202	0.0158604	0.0155881	0.0155836

Отримані значення MinHash відстані свідчать про те, що при значеннях $k \geq 13$, точність оцінки значно зростає і (з невеликими відхиленнями в межах декількох відсотків) відповідає як результатам алгоритму обчислення k-мер відстані, так і даним, отриманим іншими методами порівняння.

Відхилення можуть бути зумовлені спрощеним представленням послідовності у вигляді компактного набору мінімальних гешів.

Результати узгоджуються з тими, що отримані за допомогою інструменту з [66]. Основною перевагою даного алгоритму, попри незначне зниження точності через використання мінімальних гешів, є підвищена швидкодія та ефективніше використання ресурсів, оскільки зберігається лише невеликий набір гешів для кожної послідовності.

4.3.4 Порівняння ВП методом MinHash

Для порівняння двійкових послідовностей застосовано ті ж самі дані, що й у табл. 4.3. Результати обчислення MinHash відстані наведено у табл. 4.5, де поряд з відстанню вказано й попередньо обчислений індекс Жаккара.

Таблиця 4.5 – Обчислення MinHash відстані для випадкових послідовностей

П1	П2	MinHash відстань (попередньо порахований індекс Жакарра)				
		k=29	*k*=31	*k*=33	*k*=35	*k*=37
2MB_RandSeq1	2MB_RandSeq2	0.123774 (0.014)	0.148715 (0.005)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
6MB_RandSeq1	6MB_RandSeq2	0.086830 (0.042)	0.120698 (0.012)	0.150465 (0.0035)	1.00 (0.00)	1.00 (0.00)
13MB_RandSeq1	13MB_RandSeq2	0.062814 (0.088)	0.115789 (0.014)	0.118848 (0.010)	0.177876 (0.00099)	1.00 (0.00)

Отримані результати демонструють, що при використанні оптимально підібраної довжини k-мерів схожість ВП оцінюється приблизно у 82–85 %, хоча індекс Жаккара практично дорівнює нулю (від 0.00099 до 0.005). Таке співвідношення може свідчити про недостатню чутливість алгоритму при роботі з двійковим алфавітом. Зокрема, навіть невелике збільшення розміру k-мерів може призводити до відстані, що дорівнює 100 %, що свідчить про повну відмінність послідовностей.

4.4 Висновки до розділу

Представлені вдосконалені алгоритми, що базуються на обчисленні k-мер та MinHash відстаней, забезпечують не лише точну оцінку схожості послідовностей, а й демонструють високий рівень практичної ефективності. Особливістю алгоритму, заснованого на k-мер підході, є зниження витрат апаратних ресурсів, зокрема пам’яті, завдяки використанню оптимізованих типів даних і структур зберігання. Це дає змогу обробляти набори послідовностей на порядок більшої потужності.

Алгоритм MinHash, у свою чергу, вирізняється здатністю ефективно працювати із значно більшими розмірами порівнюваних послідовностей завдяки застосуванню геш-функції з розширеним вихідним розміром. Це відкриває можливість коректного аналізу даних на збільшених розмірностях k-мерів без різкого зростання часових витрат, що підтверджено експериментальним

порівнянням довгих послідовностей згенерованих на основі подібних ДНК-фрагментів.

Водночас, попри досягнуту високу точність обчислення схожості ДНК послідовностей, оцінка подібності двійкових послідовностей із використанням MinHash потребує подальших досліджень. Отримані результати вказують, що коректність метрики для бітових рядків може залежати від вибору функції гешування та параметрів її налаштування, тому додаткова оптимізація та адаптація під специфіку двійкових даних є перспективним напрямом роботи.

Аналіз результатів порівняння ВП та ПВП із застосуванням k-мер відстані показує, що при оптимальному виборі значення k подібність є мінімальною, що вказує на унікальність досліджуваних послідовностей і підтверджує коректність обраної стратегії визначення цього параметра. У подальшому доцільно розглянути адаптивний вибір k залежно від статистичних характеристик порівнюваних послідовностей, а також комбінування k-мер та MinHash підходів для покращення точності метрик при роботі з послідовностями різних розмірностей.

ВИСНОВКИ

У дипломній роботі виконано повний цикл досліджень, спрямований на підвищення якості генерування, аналізу та оцінки ВП і ПВП. У межах цього було послідовно проведено огляд сучасних підходів, сформульовано чіткі вимоги до випадковості, розроблено та реалізовано відповідні алгоритми, після чого здійснено їх експериментальне випробування на різноманітних наборах даних, що забезпечило відтворюваність і практичну вагомість отриманих результатів.

Проведений аналіз показав, що поєднання стандартних статистичних тестів у поєднанні з розрахунком ентропійних метрик дає цілісну та об’єктивну картину випадковості: статистичні й стохастичні підходи взаємно доповнюють одне одного і надають можливості більш глибинної оцінки генерації, адже стохастичний аналіз дозволяє оцінювати початкову ентропію джерел випадковості, а не лише вже згенеровані ВП та ПВП. Це узгоджується з нинішньою світовою практикою атестації ГВЧ і підтверджує актуальність розробленої методики.

Було системно проаналізовано ДНК-послідовності як потенційне джерело випадковості. Результати свідчать, що необроблені генетичні дані не відповідають криптографічним критеріям: спостерігаються значні зміщення частот символів та кореляційні залежності, які знижують рівень ентропії. Отже, без додаткової пост-обробки ДНК-послідовності не можуть слугувати криптографічно стійким джерелом випадкових бітів. Однак, розроблені та запропоновані алгоритми генерації ВП та ПВП, що використовують екстракцію та випадкові сеансові параметри, виявилися здатними генерувати послідовності, які відповідають критеріям як статистичного, так і стохастичного тестувань, підтверджуючи практичну придатність запропонованого підходу.

З технічного погляду також було створено й випробувано вдосконалені алгоритми оцінки подібності послідовностей. Оптимізований k-мерний метод забезпечив помітне скорочення використання пам’яті під час порівняння, а розширений алгоритм MinHash підвищив точність та ефективність вимірювання

схожості при роботі з послідовностями великого розміру. Експериментальні випробування показали, що навіть за оптимально підібраних параметрів k-мерного аналізу згенеровані запропонованими алгоритмами послідовності мають майже нульову подібність, що підкреслює їхню унікальність. Разом із тим точність MinHash виявила чутливість до використовуваних параметрів гешування – цей аспект потребує подальших удосконалень для підвищення стабільності результатів.

Подальші дослідження доцільно спрямувати на вдосконалення алгоритмів порівняння та більш ретельну оцінка розробленого ГВЧ на основі ДНК, наприклад, доведення характеристик прямої та зворотної секретності та покращених прямої та зворотної секретності, що має суттєве значення для інтеграції у сучасні криптографічні протоколи.

Таким чином, мету роботи досягнуто: запропоновано нові методи генерації ВП та ПВП на основі ДНК та екстракції, їх статистичної та стохастичної оцінки, а також алгоритми порівняння, що відкриває можливості для подальших наукових і практичних розробок у цьому напрямку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Голубничий Д. Ю., Кандій С. О., Єсіна М. В., Горбенко Д. Ю. Методи та засоби аналізу, оцінки та порівняння властивостей випадкових послідовностей та випадкових чисел [Електронний ресурс] // *Радіотехніка : всеукр. міжвідом. наук.-техн. зб.* – 2024. – Вип. 216. – С. 30–45. – Режим доступу: http://rt.nure.ua/issue/view/18026/10971.

2. Stipčević M., Koç Ç. K. True Random Number Generators // *Open Problems in Mathematics and Computational Science*. – Berlin : Springer, 2014. – pp. 275–315.

3. Mannalatha V., Mishra S., Pathak A. A Comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness [Електронний ресурс]. – 2022. – Режим доступу: https://arxiv.org/pdf/2203.00261.pdf.

4. Beenakker C. W. J., Büttiker M. Suppression of shot noise in metallic diffusive conductors // *Physical Review B*. – 1992. – Vol. 46. – pp. 1889–1892.

5. Nyquist H. Thermal agitation of electric charge in conductors // *Physical Review*. – 1928. – Vol. 32. – pp. 110–113.

6. Guo H., Tang W., Liu Y., Wei W. Truly random number generation based on measurement of phase noise of a laser // *Physical Review E*. – 2010. – Vol. 81. – Art. 051137.

7. Li X., Cohen A. B., Murphy T. E., Roy R.Scalable parallel physical random number generator based on a superluminescent LED // *Optics Letters*. – 2011. – Vol. 36, № 5. – pp. 1020–1022.

8. Bagini V., Bucci M. A design of reliable true random number generator for cryptographic applications // *Cryptographic Hardware and Embedded Systems – CHES 2002 : Proc. Int. Conf.* – Berlin : Springer, 2002. – pp. 204–218.

9. Stipčević M. Fast nondeterministic random bit generator based on weakly correlated physical events // *Review of Scientific Instruments*. – 2004. – Vol. 75. – pp. 4442–4449.

10. Stipčević M. Apparatus and method for generating true random bits based on time integration of an electronic noise source : Int. Patent WO 03/040854 A1. – 2003. – 9 p.

11. von Neumann J. Various techniques for use in connection with random digits // *John von Neumann Collected Works*. – 1963. – Vol. 5. – pp. 768–770.

12. Kanter I., Aviad Y., Reidler I., Cohen E., Rosenbluh M. An optical ultrafast random bit generator // *Nature Photonics*. – 2010. – Vol. 4, № 1. – pp. 58–61.

13. Wang A. B., Wang Y. C., He H. C. Enhancing the bandwidth of the optical chaotic signal generated by a semiconductor laser with optical feedback // *IEEE Photonics Technology Letters*. – 2008. – Vol. 20, № 19. – pp. 1633–1635.

14. Reidler I., Aviad Y., Rosenbluh M., Kanter I. Ultra-high-speed random number generation based on a chaotic semiconductor laser // *Physical Review Letters*. – 2009. – Vol. 103, № 2. – Art. 024102.

15. Wang A. B., Wang Y. C., Wang J. F. Route to broadband chaos in a chaotic laser diode subject to optical injection // *Optics Letters*. – 2009. – Vol. 34, № 8. – pp. 1144–1146.

16. Rodriguez-Henriquez F., Saqib N. A., Diaz-Perez A., Koç Ç. K. *Cryptographic Algorithms on Reconfigurable Hardware*. – Berlin : Springer, 2007. – 280 p.

17. Dichtl M., Golić J. D. High-speed true random number generation with logic gates only // *Cryptographic Hardware and Embedded Systems – CHES 2007 : Proc. Int. Conf.* – Berlin : Springer, 2007. – pp. 45–62.

18. Schmidt H. Quantum-Mechanical Random-Number Generator // *Journal of Applied Physics*. – 1970. – Vol. 41, № 2. – pp. 462–468.

19. Vincent C. H. The generation of truly random binary numbers // *Journal of Physics E: Scientific Instruments*. – 1970. – Vol. 3, № 8. – pp. 594–598.

20. Pironio S., Acín A., Massar S. et al. Random numbers certified by Bell's theorem // *Nature*. – 2010. – Vol. 464, № 7291. – pp. 1021–1024.

21. Um M., Zhang X., Zhang J. et al. Experimental certification of random numbers via quantum contextuality // *Scientific Reports*. – 2013. – Vol. 3, Art. 1627. – pp. 1–6.

22. Katsoprinakis G. E., Polis M., Tavernarakis A. et al. Quantum random number generator based on spin noise // *Physical Review A*. – 2008. – Vol. 77. – Art. 054101.

23. Jennewein T., Achleitner U., Weihs G., Weinfurter H., Zeilinger A. A fast and compact quantum random number generator // *Review of Scientific Instruments*. – 2000. – Vol. 71, № 4. – pp. 1675–1680.

24. Weihs G., Jennewein T., Simon C., Weinfurter H., Zeilinger A. Violation of Bell's inequality under strict Einstein locality conditions // *Physical Review Letters*. – 1998. – Vol. 81. – pp. 5039–5043.

25. Wang P. X., Long G. L., Li Y. S. Scheme for a quantum random number generator // *Journal of Applied Physics*. – 2006. – Vol. 100, № 5. – Art. 056107.

26. IBM Quantum [Електронний ресурс]. – Режим доступу: https://quantum-computing.ibm.com.

27. Stefanov A., Gisin N., Guinnard O., Guinnard L., Zbinden H. Optical quantum random number generator // *Journal of Modern Optics*. – 2000. – Vol. 47, № 4. – pp. 595–598.

28. Grafe M., Heilmann R., Perez-Leija A. et al. On-chip generation of high-order single-photon W-states // *Nature Photonics*. – 2014. – Vol. 8, № 10. – pp. 791–795.

29. *Security Requirements for Cryptographic Modules* (FIPS 140-1) [Електронний ресурс]. – 1994. – Режим доступу: https://csrc.nist.gov/csrc/media/Publications/fips/140/1/archive/1994-01-11/documents/fips1401.pdf.

30. *Security Requirements for Cryptographic Modules* (FIPS 140-2) [Електронний ресурс]. – 2002. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf.

31. *Security Requirements for Cryptographic Modules* (FIPS 140-3) [Електронний ресурс]. – 2019. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-3.pdf.

32. Marsaglia G. *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness* [Електронний ресурс]. – Режим доступу: http://stat.fsu.edu/pub/diehard/.

33. Rukhin A. A. *Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. NIST SP 800-22 rev 1a [Електронний ресурс]. – 2010. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf.

34. Гріненко Т. О., Горбенко І. Д. Методики та засоби тестування послідовностей випадкових чисел та особливості їх застосування // *Прикладная радиоэлектроника*. – 2006. – Т. 5, № 1. – С. 115–126.

35. Brown R. G. *DieHarder: A GNU Public License Random Number Tester*. – Durham (NC) : Duke Univ., 2005. – 43 p.

36. Brown R. G., Eddelbuettel D., Bauer D. *Dieharder: A Random Number Test Suite. Ver. 3.31.1* [Електронний ресурс]. – Режим доступу: https://webhome.phy.duke.edu/~rgb/General/dieharder.php.

37. *BSI AIS 31. A Proposal for Functionality Classes for Random Number Generators* [Електронний ресурс]. – Sept. 2022. – Режим доступу: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Certification/Interpretations/AIS_31_Functionality_classes_for_random_number_generators_e.pdf.

38. Turan M., Barker E., Kelsey J. et al. *NIST SP 800-90B. Recommendation for the Entropy Sources Used for Random Bit Generation* [Електронний ресурс]. – 2018. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf.

39. Реньї А. Про міри інформації та ентропії // *Труди IV Берклiївського симпозіуму з математики, статистики та теорії ймовірностей*. – 1960. – С. 547.

40. Закон України «Про електронні довірчі послуги» від 05.10.2017 № 2155 VIII (ред. 01.01.2024) [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2155-19.

41. ISO/IEC 20543:2019 *Information technology – Security techniques – Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408*. – Geneva : ISO, 2019.

42. *Common Methodology for IT Security Evaluation. Evaluation Methodology. Version 3.1, Revision 4* [Електронний ресурс]. – Sept. 2012. – Режим доступу: https://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R4.pdf.

43. *Common Methodology for IT Security Evaluation. Evaluation Methodology. Version 2.3* [Електронний ресурс]. – Aug. 2005. – Режим доступу: https://www.commoncriteriaportal.org/files/ccfiles/cemv2.3.pdf.

44. Rodríguez L., Madarro-Capó E., Legón-Pérez C. et al. Selecting an Effective Entropy Estimator for Short Sequences of Bits // *Entropy*. – 2021. – Vol. 23, № 5. – Art. 561.

45. Skorski M. Improved estimation of collision entropy in high- and low-entropy regimes and applications to anomaly detection [Електронний ресурс]. – Режим доступу: https://eprint.iacr.org/2016/1035.pdf.

46. Paninski L. Estimation of entropy and mutual information // *Neural Computation*. – 2003. – Vol. 15, № 5. – pp. 1191–1253.

47. Trybula S. Some problems of simultaneous minimax estimation // *Annals of Mathematical Statistics*. – 1958. – Vol. 29. – pp. 245–253.

48. Hausser J., Strimmer K. Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks // *Journal of Machine Learning Research*. – 2009. – Vol. 10. – pp. 1469–1484.

49. Valiant G., Valiant P. Estimating the unseen: improved estimators for entropy and other properties // *Journal of the ACM*. – 2017. – Vol. 64, № 6. – Art. 1. – pp. 1–41.

50. Schürmann T. Bias analysis in entropy estimation // *Journal of Physics A: Mathematical and General*. – 2004. – Vol. 37. – pp. L295–L301.

51. Chao A., Shen T. J. Non-parametric estimation of Shannon's index of diversity when there are unseen species in sample // *Environmental and Ecological Statistics*. – 2003. – Vol. 10, № 5. – pp. 429–443.

52. Hagerty P., Draper T. Entropy bounds and statistical tests // *Proceedings of the NIST Random Bit Generation Workshop*, 2012. – 8 p.

53. Kelsey J., McKay K. A., Turan M. S. Predictive models for min-entropy estimation // *CHES 2015 : Proc. Int. Conf. on Cryptographic Hardware and Embedded Systems*. – LNCS 9293. – 2015. – pp. 82–99.

54. Maurer U. A universal statistical test for random bit generators // *Journal of Cryptology*. – 1992. – Vol. 5, № 2. – pp. 89–105.

55. Дерев'янко Я. А., Єсіна М. В., Горбенко Д. Ю. Обґрунтування методів обчислення та аналіз властивостей псевдовипадкових та випадкових послідовностей на основі ДНК [Електронний ресурс] // *Радіотехніка : всеукр. міжвідом. наук.-техн. зб.* – 2024. – Вип. 217. – С. 23–38. – Режим доступу: http://rt.nure.ua/issue/view/18317/11263.

56. National Center for Biotechnology Information (NCBI) [Електронний ресурс]. – Режим доступу: https://ftp.ncbi.nlm.nih.gov/genbank/.

57. DDBJ – DNA Data Bank of Japan [Електронний ресурс]. – Режим доступу: https://ddbj.nig.ac.jp/arsa/quick_search.

58. *NIST SP 800-90A Rev. 1. Recommendation for Random Number Generation Using Deterministic Random Bit Generators* [Електронний ресурс]. – 2015. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf.

59. Quantis QRNG USB [Електронний ресурс]. – Режим доступу: https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator/.

60. АТ «ІІТ». Апаратний ГВЧ «Гряда-3» [Електронний ресурс]. – Режим доступу: https://iit.com.ua/index.php?page=itemdetails&p=3&gtype=1&type=1&id=96.

61. Müller S. *Linux /dev/random – A New Approach*. – 2024. [Електронний ресурс] – Режим доступу: https://www.chronox.de/lrng/releases/v53/lrng-v53.pdf.

62. Barman P., Djilali B., Benatmane S., Nuh A. A new hybrid cryptosystem involving DNA, Rabin, One-Time Pad and Feistel // *SSRN Electronic Journal*. – 2024. [Електронний ресурс] – Режим доступу: https://doi.org/10.2139/ssrn.4771411.

63. Zhang Y., Liu X., Sun M. DNA based random key generation and management for OTP encryption // *BioSystems*. – 2017. – Vol. 159. – pp. 51–63.

64. Siddaramappa V., Ramesh K. B. True random number generation based on DNA molecule genetic information (DNA-TRNG) // *Cryptology ePrint Archive*. – 2020. – Paper 2020/745.

65. UCDavis Bioinformatics Core. *Genome Assembly Workshop* [Електронний ресурс]. – Режим доступу: https://ucdavis-bioinformatics-training.github.io/2020-Genome_Assembly_Workshop/kmers/kmers.

66. Ondov B. D., Treangen T. J., Melsted P. et al. Mash: fast genome and metagenome distance estimation using MinHash // *Genome Biology*. – 2016. – Vol. 17. – Art. 132.

67. Wilkinson S. Fast k-mer counting and clustering for biological sequence analysis [Електронний ресурс]. – 2022. – Режим доступу: https://cran.r-project.org/web/packages/kmer/kmer.pdf.

68. Edgar R. C. Local homology recognition and distance measures in linear time using compressed amino acid alphabets // *Nucleic Acids Research*. – 2004. – Vol. 32, № 1. – pp. 380–385.

69. Kelsey J. Truncation mode for SHA-256 (and most other hashes) [Електронний ресурс]. – 2005. – Режим доступу: https://csrc.nist.gov/csrc/media/events/first-cryptographic-hash-workshop/documents/kelsey_truncation.pdf.

70. Dang Q. *NIST SP 800-107. Recommendation for Applications Using Approved Hash Algorithms* [Електронний ресурс]. – 2009. – Режим доступу: https://csrc.nist.rip/library/NIST%20SP%20800-107%20Recommendation%20for%20Apps%20Using%20Approved%20Hash%20Algorithms,%202009-02.pdf.

ДОДАТОК А

ДШ, ЩО МОЖУТЬ ЗАСТОСОВУВАТИСЯ ДЛЯ ГЕНЕРАЦІЇ ВП НА ОСНОВІ PTRNG

Таблиця А.1 – Фізично справжні ДШ

Тип	Переваги	Недоліки	Особливості
На основі часу проходження фотонів.	Генерація бітів відбувається зі швидкістю до 10^9 Мбіт/с завдяки вимірюванню часових інтервалів між окремими фотонами. Комбінування аналізу часу прибуття та підрахунку тактових циклів дозволяє підвищити рівень ентропії, мінімізуючи вплив випадкових затримок і підвищуючи стійкість до надмірних повторів.	Для роботи потрібні надточні таймери та висока стабільність генераторів тактових імпульсів, оскільки похибки в реєстрації часу призводять до кореляції між вихідними бітами. Висока складність калібрування й обмежена можливість масштабування через необхідність синхронізації багатьох фотодетекторів.	Використовують слабкі фотонні джерела (лазер або СД) і швидкі фотодетектори, які фіксують час прибуття фотонів. Експоненційне перетворення інтервалів дозволяє вирівняти початковий розподіл, а адаптивне порівняння з тактовими імпульсами забезпечує формування вихідних бітів при запиті системи.

Продовження таблиці А.1

Тип	Переваги	Недоліки	Особливості
На багатофотонних детекторах	Одне вимірювання може генерувати до 1,99 біта випадковості, що підвищує пропускну здатність до ~144 Мбіт/с. Завдяки можливості використовувати слабкі когерентні джерела та твердотільні однофотонні випромінювачі, систему можна інтегрувати навіть у мобільні пристрої з обмеженими ресурсами.	Потрібна значна постобробка для вирівнювання розподілу та видалення залишкових кореляцій. Щоб зберегти статистичну однорідність, середня кількість фотонів на подію має бути обмежена (< 0,1), інакше зростає ризик мультифотонних імпульсів та спотворення ентропії.	Фіксується кількість фотонів у когерентному стані відповідно до розподілу Пуассона. Використовуються методи підрахунку імпульсів або визначення інтервалів між ними. Умонтовані RC-ланцюги та адаптивні фільтри забезпечують попереднє вирівнювання експоненційного розподілу, перш ніж передати дані в механізми корекції помилок.
На основі макроскопічного фотодетектування	Оминає “мертвий час” однофотонних детекторів, оскільки генерує випадковість із	Потрібно забезпечити домінування квантових флуктуацій над класичним шумом	Використовується гомодинна схема з двома фотодетекторами: різницеве вимірювання

Продовження таблиці А.1

Тип	Переваги	Недоліки	Особливості
	класичних параметрів світла (інтенсивності й амплітуди). Завдяки цьому система досягає дуже високих швидкостей генерації без необхідності очікування на перезарядку фотодетекторів.	світла, що вимагає складної збалансованої гомодинної схеми. Невеликі розбалансування між каналами виявлення призводять до зростання кореляцій та зниження якості випадковості.	компенсує класифікаційний шум гомодину, а подальша цифрова обробка дозволяє отримати рівномірно розподілені біти. Підходить для каналів із низькими втратами та може бути адаптована для систем квантового розподілу ключів.
На основі вакуумного шуму	Забезпечує фундаментальну випадковість, оскільки використовує вакуумні флуктуації електромагнітного поля, які не залежать від класичних шумів. Системи досягають швидкості до 3 Гбіт/с при	Чутливі до дробового шуму детекторів та електромагнітних перешкод, тому потребують високопродуктивних фотодетекторів і надточного балансування каналів. Будь-яка нестабільність у потужності лазера або розсіювачів призводить до.	Лазерний імпульс проходить через розсіювач променя, після чого два детектори виконують балансоване виявлення вакуумного стану. Різницеве вимірювання між каналами виділяє квантові флуктуації. Застосовується синхронний високочастотний

Продовження таблиці А.1

Тип	Переваги	Недоліки	Особливості
	збалансованому гомодинному виявленні, що робить їх одними із найшвидших QRNG.	зміщення статистики флуктуацій	підсилювач і цифрова обробка для перетворення аналогового сигналу на бітову послідовність.
На основі ефекту посиленого спонтанного випромінювання	Використання фазового шуму ASE дозволяє уникнути обмежень дробового шуму та підвищити швидкість генерування бітів у порівнянні з вакуумними схемами. Системи на цій основі демонструють високу ефективність при використанні існуючих лазерних та гетеродинних конфігурацій.	Для стабільної роботи необхідно точно контролювати фазу лазерних променів і стабілізувати довжину інтерферометра, що збільшує вартість і складність реалізації. Будь-які флуктуації в температурі або живленні можуть спричинити зсув статистики випадковості.	Генерація здійснюється шляхом інтерференції слабких когерентних станів у гетеродинному режимі. Вимірювання квадратур циклічно фазово рандомізованих імпульсів дає змогу отримувати випадкові біти. Використовуються пікові імпульсні лазерні джерела та цифрова фільтрація для усунення залишкових кореляцій.

Кінець таблиці А.1

Тип	Переваги	Недоліки	Особливості
На основі Раманівського розсіювання	Застосування спонтанного та вимушеного комбінаційного розсіювання фотонів забезпечує високу якість випадковості завдяки квантовим флуктуаціям під час взаємодії з молекулярними ґратами. Можливість вибору між фазовим і амплітудним режимами робить цю технологію гнучкою для різноманітних застосувань.	Системи потребують дорогого оптичного обладнання високої точності та мають обмеження за швидкістю через оптичні втрати та кореляції в середовищі. Точне вирівнювання і калібрування компонентів є складним і чутливим до зовнішніх умов (температури, вібрацій тощо).	Використовують схему, показану на рис. 1.11: фотони проходять через молекулярну систему, де виникає спонтанне (SpRS) або стимульоване (SRS) комбінаційне розсіювання. Інтенсивність або фаза розсіяного сигналу аналізується інтерферометричними методами, а цифрова обробка видаляє залишкові кореляції й забезпечує рівномірний розподіл бітів.

ДОДАТОК Б

ПЕРЕЛІК СТАТИСТИЧНИХ ТЕСТІВ НАБОРУ DIEHARD

1) Тест днів народження (Birthday Spacings Test) – аналізує інтервали між випадковими значеннями, порівнюючи їх із розподілом Пуассона.

2) Тест п’яти пересічних перестановок (Overlapping 5-Permutation Test) – перевіряє частотність можливих впорядкувань підпослідовностей у великих наборах випадкових чисел.

3) Тест рангів матриць (Binary Rank Test for 31×31 matrices) – створює 31×31 двійкові матриці та аналізує їхні ранги.

4) Тест рангів матриць (Binary Rank Test for 32×32 matrices) – аналогічний попередньому, але для 32×32 матриць.

5) Тест рангів матриць (Binary Rank Test for 6×8 matrices) – працює з 6×8 матрицями, оцінюючи їхній розподіл рангів.

6) Потоковий тест (Bitstream Test) – аналізує 20-бітні підпослідовності у бітовому потоці.

7) Тест OPSO (Overlapping-Pairs Sparse Occupancy Test) – перевіряє частоту появи 2-літерних слів у послідовності.

8) Тест OQSO (Overlapping-Quadruples Sparse Occupancy Test) – аналогічний OPSO, але працює з 4-літерними словами.

9) Тест DNA – оцінює появу 10-літерних слів у послідовності за принципом ДНК-кодування.

10) Тест підрахунку одиниць у потоці байтів (Count-the-1’s Test on a stream of bytes) – перевіряє частоту появи 1 у всіх байтах послідовності.

11) Тест підрахунку одиниць для конкретних байтів (Count-the-1’s Test for specific bytes) – аналогічний попередньому, але працює з вибраними байтами.

12) Тест на розташування (Parking Lot Test) – моделює випадкове розміщення об’єктів у площині та аналізує закономірності.

13) Тест на мінімальну відстань (Minimum Distance Test) – визначає найменшу відстань між випадковими точками в площині, використовуючи експоненційний розподіл.

14) Тест об'ємних сфер (3D-Spheres Test) – перевіряє розподіл сфер у просторі, обчислюючи їхні мінімальні радіуси.

15) Тест гри у кістки (Craps Test) – моделює 200 000 ігор у кості, підраховуючи кількість перемог і середню кількість кидків до завершення гри.

ДОДАТОК В

ПЕРЕЛІК ВИМОГ СТАНДАРТУ NIST SP800-90B ДО ДШ

1) Наявність обґрунтованої стохастичної моделі вихідних сигналів ДШ. Модель повинна включати опис того, як працює ДШ та яким чином створюється непередбачуваність, а також і обґрунтування того, чому джерело шуму забезпечує прийнятну вихідну ентропію.

2) Поведінка джерела шуму має бути стаціонарною, коли розподіли ймовірностей вихідних сигналів джерела шуму не змінюються з часом при роботі джерела в нормальних умовах. Для цього повинне бути обґрунтовано, звідки походить непередбачуваність, і приблизно описано поведінку ДШ щодо стаціонарності його поведінки.

3) Модель ДШ повинна надавати чітке визначення очікуваної ентропії, що забезпечується вихідними сигналами джерела шуму, і надавати технічну аргументацію, чому джерело шуму може підтримувати таку швидкість ентропії.

4) Стан джерела шуму має бути максимально захищений від впливу. Методи, що використовуються для цього, повинні бути задокументовані, щодо межі безпеки захисту ДШ від впливу.

5) Незважаючи на те, що джерело шуму не зобов’язане створювати неупереджені та незалежні вихідні сигнали, воно повинно демонструвати випадкову поведінку, коли вихід не може бути визначений жодним відомим алгоритмічним правилом.

6) Джерело шуму має генерувати випадкові значення фіксованої довжини та має опис вихідного простору джерела шуму.

7) Якщо для підвищення безпеки використовуються додаткові ДШ, необхідно мати документ, який описує додаткові джерела шуму.

ДОДАТОК Г

АЛГОРИТМИ ГЕНЕРАЦІЇ ПВП ТА ВП НА ОСНОВІ ДНК

Алгоритм генерації ПВП ComputePRS:

Вхід: Файл із ДНК-фрагментом (бінарний формат, довжина 1 МБ); Ключ (HEX), 256 біт; Nonce (HEX), 256 біт.

Вихід: Файл із ПВП (бінарний формат, розміром кратний 1 МБ) на основі заданої ДНК.

Послідовність дій:

1) Задання розміру ключа, nonce та відкритого/зашифрованого тексту (у байтах) та початкова ініціалізація блокового шифру

#define KEY_SIZE 32;

#define NONCE_SIZE 32;

#define PT_SIZE 32;

#define CT_SIZE 32;

ctx = KalynaInit;

2) Ініціалізація буферу, в який буде зчитуватися послідовність ДНК (фрагмент необхідного розміру, наприклад, 1МБ)

rawSeq[1 MB/8];

3) Зчитування ДНК послідовності у двійковому вигляді

rawSeq[i] = байти[i – i+8];

4) Обчислення послідовності обраної довжини

while (обраної довжини послідовності не досягнуто)

Ініціалізація ключа та nonce

if (перша ітерація зашифрування)

Задання ключа і nonce через вхідні дані

else

initKey = внутрішній стан генератора;

nonce = внутрішній стан генератора;

uint64_t initKey [KEY_SIZE/8] = Hash32(initKey | 0x01);

```
        uint64_t nonce [NONCE_SIZE/8] = Hash32(nonce | 0x02);
    end if
    Зашифрування початкового/уже зашифрованого фрагменту
    KalynaKeyExpand;
    counter = 0;
        for (i < rawSeqLen; i+= PT_SIZE/8)
            pt [PT_SIZE/8] = {nonce[i] … nonce[i+PT_SIZE/8-1]};
            pt = pt^counter;
            counter++;
            KalynaEncipher(pt, ctx, ct);
            ct[i] … ct[i+CT_SIZE /8-1]  = ct[i…i+…]^rawSeq[i…i+…];
            rawSeq [i] … rawSeq [i+CT_SIZE/8-1] = ct[i] … ct[i+…];
        end for
        Запис зашифрованого фрагменту у вихідний файл
end while.
```

Алгоритм генерації ВП ComputeRS:

Вхід: Файл із бінарним ДНК-фрагментом – довжина 1 МБ.

Вихід: Файл із випадковою послідовністю на основі заданої ДНК (бінарний формат, розміром кратний 1 МБ).

Послідовність дій:

1) Задання розміру ключа, nonce та відкритого/зашифрованого тексту (у байтах) та початкова ініціалізація блокового шифру

```
#define KEY_SIZE 32;
#define NONCE_SIZE 32;
#define PT_SIZE 32;
#define CT_SIZE 32;
ctx = KalynaInit;
```

2) Ініціалізація буферу, в який буде зчитуватися послідовність ДНК (фрагмент необхідного розміру, наприклад 1 МБ)

```
rawSeq[1 MB/8];
3) Зчитування ДНК послідовності у двійковому вигляді
rawSeq[i] = байти[i – i+8];
4) Обчислення послідовності обраної довжини
while (обраної довжини послідовності не досягнуто)
  Ініціалізація ключа та nonce
  file = open("/dev/random", "r");
  read(file, initKey, KEY_SIZE);
  read(file, nonce, NONCE_SIZE);
  Зашифрування початкового/уже зашифрованого фрагменту
  KalynaKeyExpand;
  counter = 0;
        for (i < rawSeq Len; i+= PT_SIZE/8)
              pt[PT_SIZE/8] = {nonce[i] … nonce[i+PT_SIZE/8-1]};
              pt = pt^counter;
              counter++;
              KalynaEncipher(pt, ctx, ct);
              ct[i] … ct[i+PT_SIZE/8-1]  = ct[i…i+…]^rawSeq[i…i+…];
              rawSeq [i] … rawSeq [i+PT_SIZE/8-1] = ct[i] … ct[i+…];
        end for
        Запис у файл зашифрованої частини
  end while.
```

ДОДАТОК Д

СТАТИСТИЧНІ ПОРТРЕТИ ЗГЕНЕРОВАНИХ ПВП ТА ВП

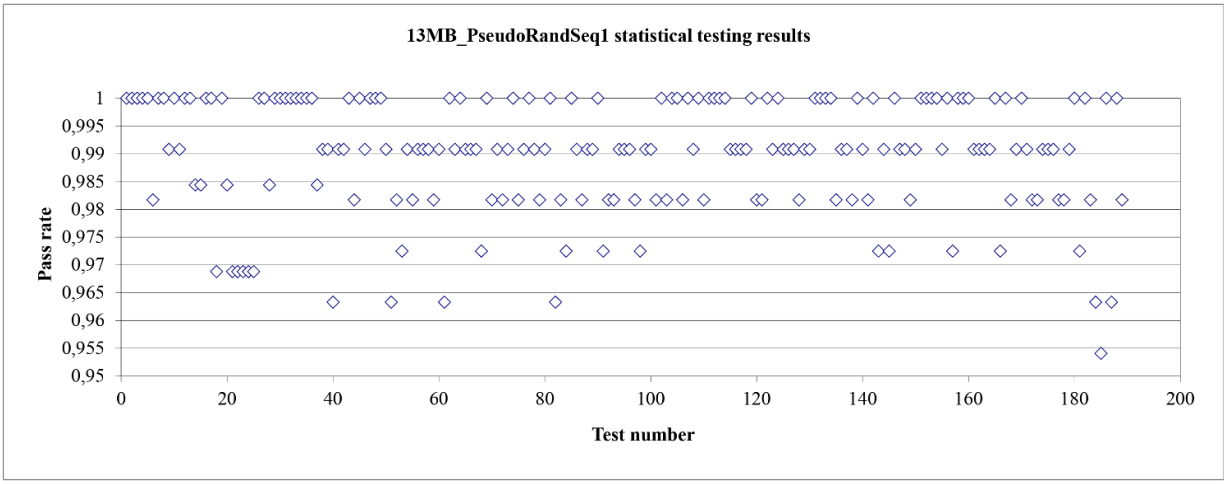

Рисунок Д.1 – Статистичний портрет ПВП 13MB_PseudoRandSeq1

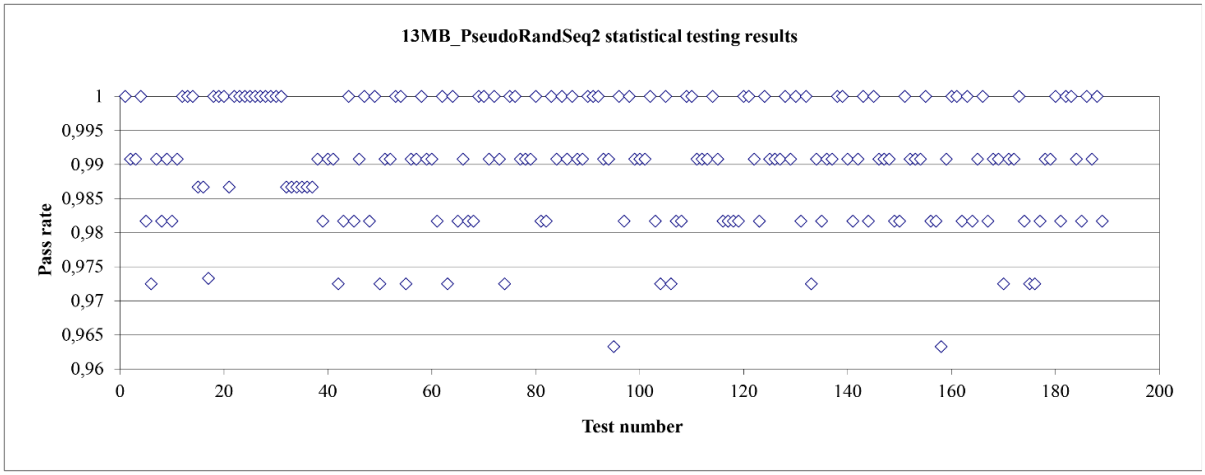

Рисунок Д.2 – Статистичний портрет ПВП 13MB_PseudoRandSeq2

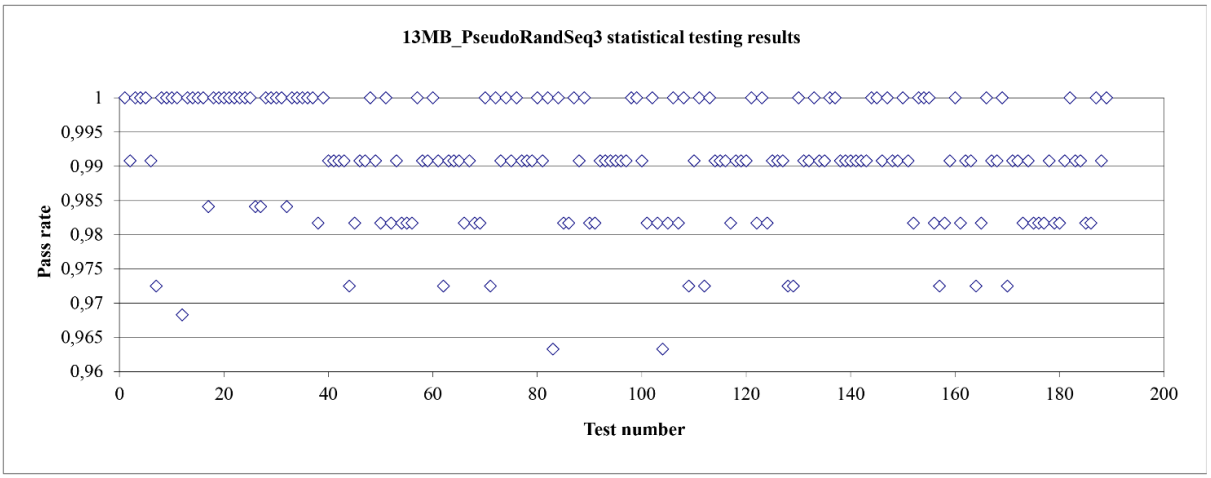

Рисунок Д.3 – Статистичний портрет ПВП 13MB_PseudoRandSeq3

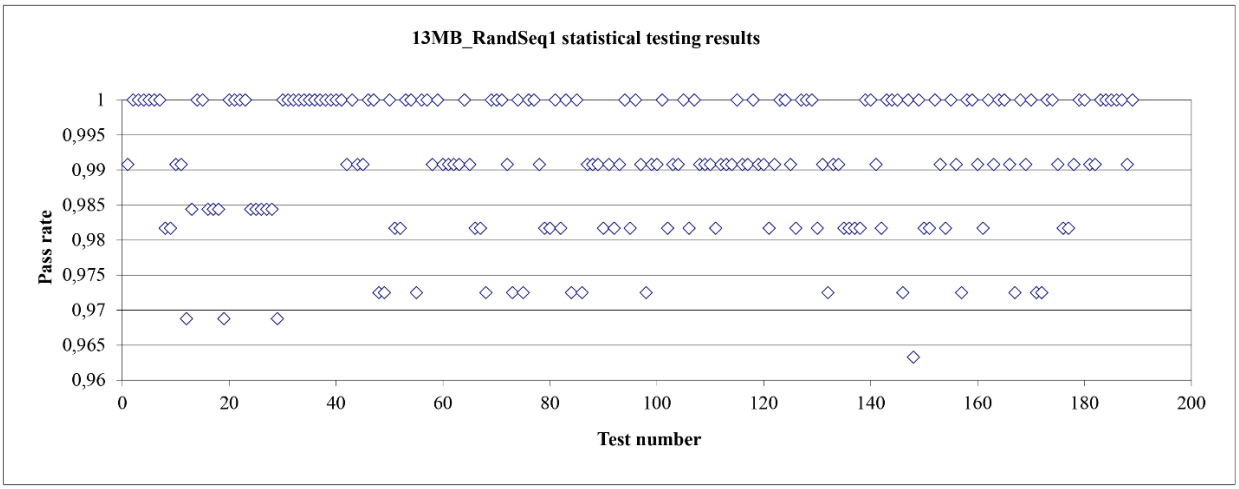

Рисунок Д.4 – Статистичний портрет ВП 13MB_RandSeq1

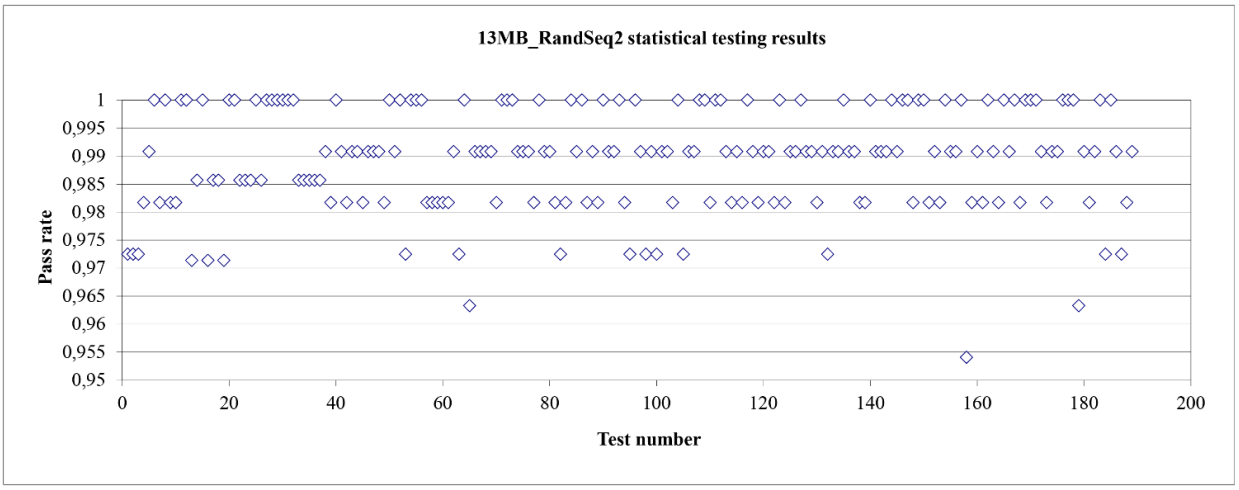

Рисунок Д.5 – Статистичний портрет ВП 13MB_RandSeq2

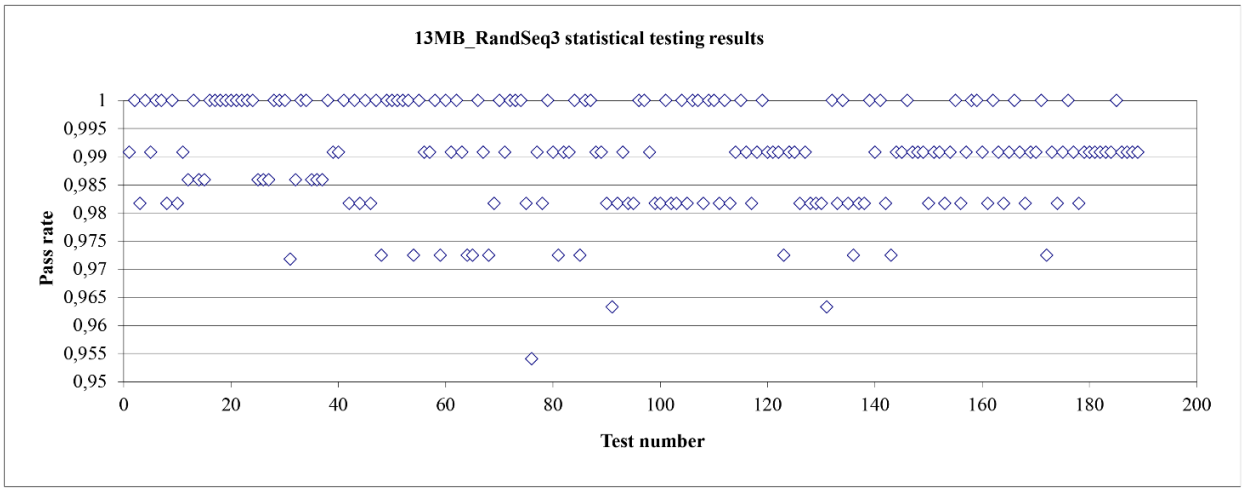

Рисунок Д.6 – Статистичний портрет ВП 13MB_RandSeq3

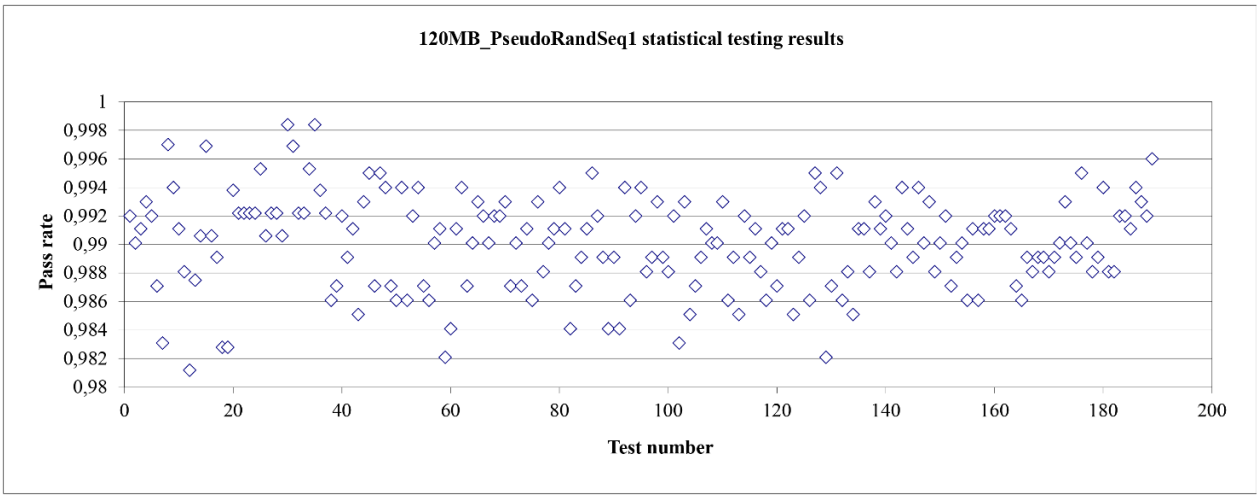

Рисунок Д.7 – Статистичний портрет ПВП 120MB_PseudoRandSeq1

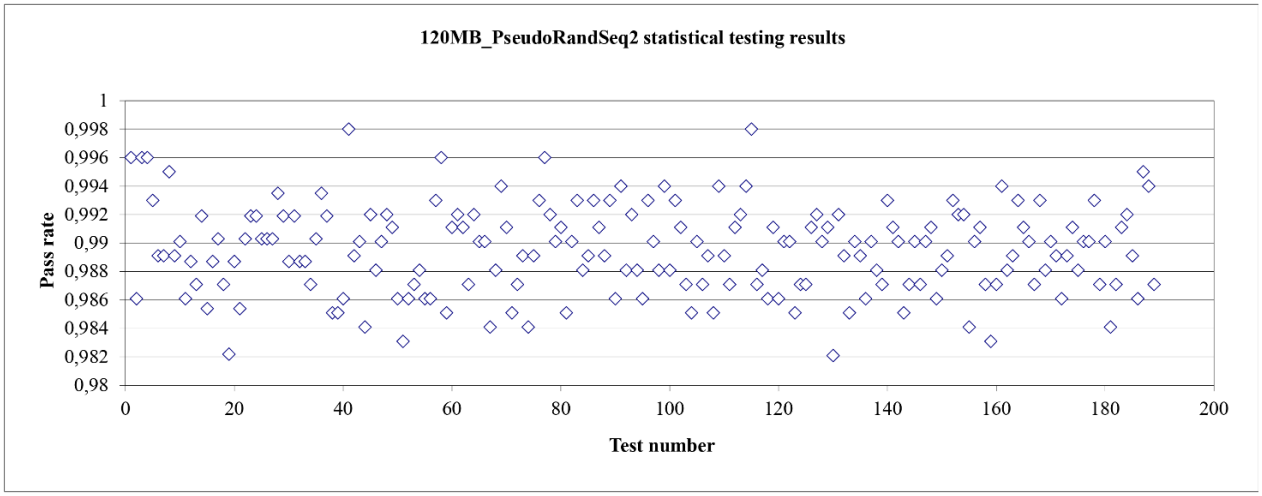

Рисунок Д.8 – Статистичний портрет ПВП 120MB_PseudoRandSeq2

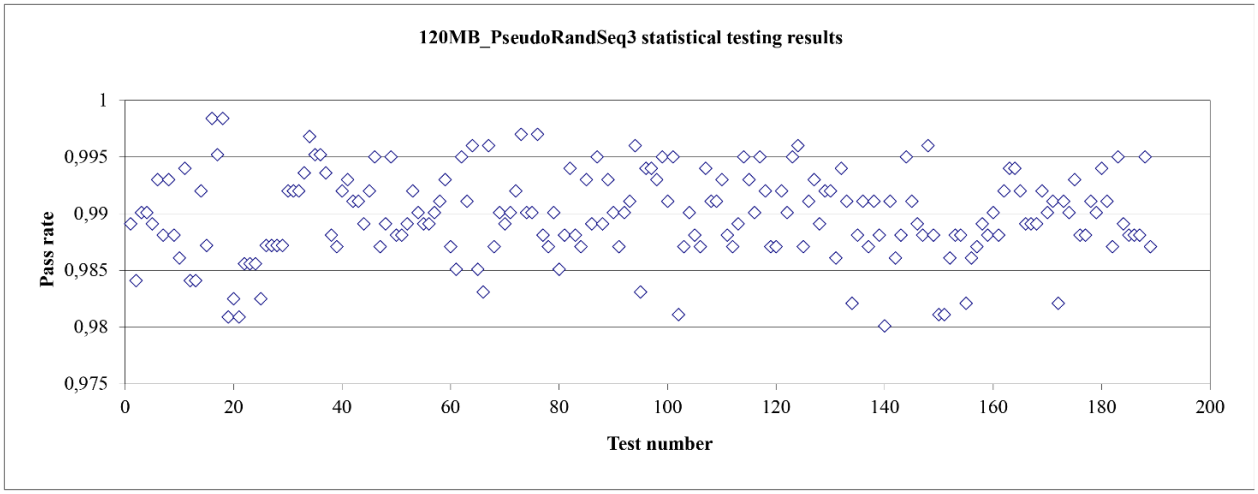

Рисунок Д.9 – Статистичний портрет ПВП 120MB_PseudoRandSeq3

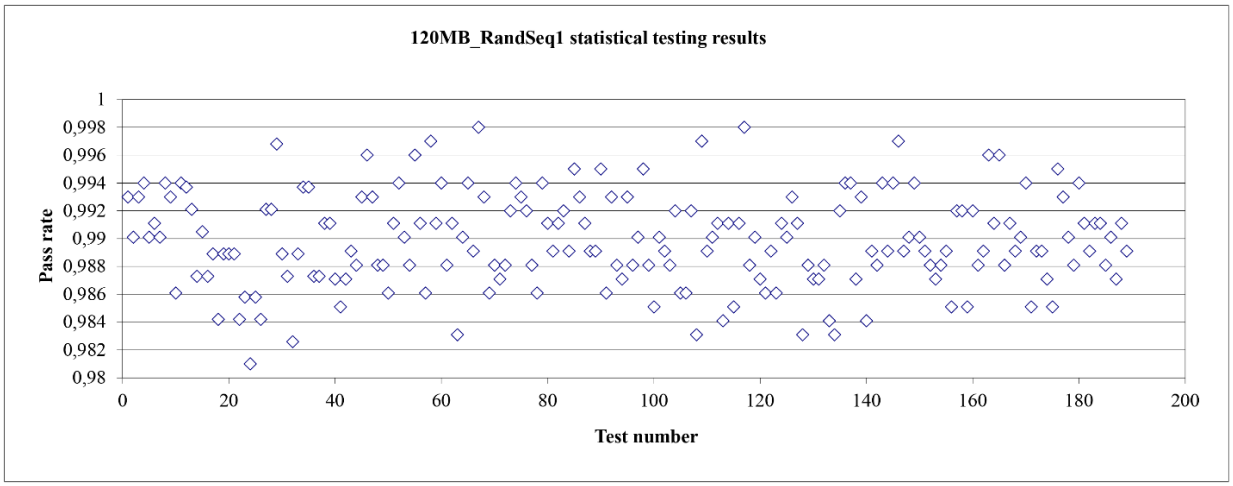

Рисунок Д.10 – Статистичний портрет ВП 120MB_RandSeq1

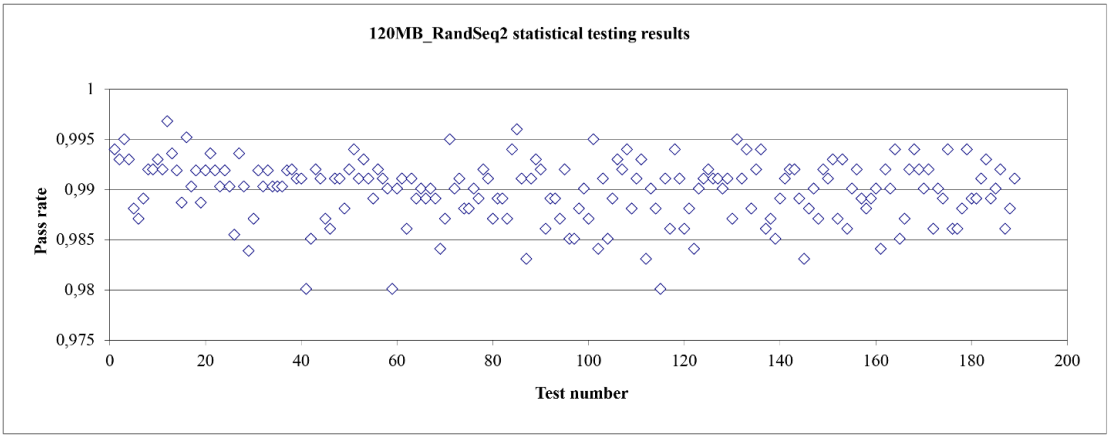

Рисунок Д.11 – Статистичний портрет ВП 120MB_RandSeq2

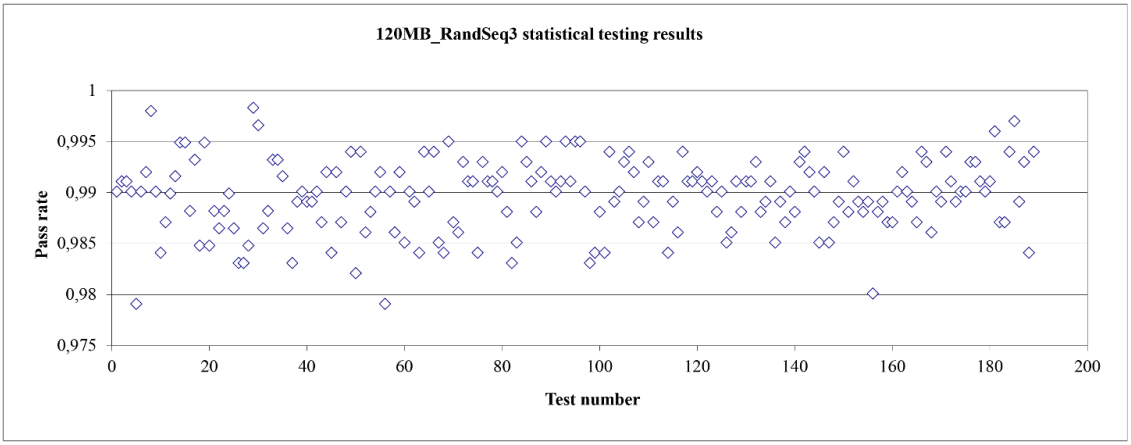

Рисунок Д.12 – Статистичний портрет ВП 120MB_RandSeq3

ДОДАТОК Е

РОЗРОБЛЕНІ АЛГОРИТМИ ОЦІНКИ ПОДІБНОСТІ ПОСЛІДОВНОСТЕЙ

Алгоритм на основі обчислення k-мер відстані KmerDistCount:

Вхід: Дві послідовності (ДНК, ПВП, ВП – у відповідних текстових чи бінарних форматах, таких як .fasta, .fna, .bin, .dat тощо), для яких необхідно оцінити схожість.

Вихід: Обчислена k-мер відстань, що характеризує ступінь схожості.

Послідовність дій:

1) Розбиття обох послідовностей на k-мери (довжина k визначається згідно з формулою (4.1) або (4.2)).

2) Перетворення k-мерів у двійкове представлення згідно з визначеними правилами (див. розділ 3) та конвертація в 64-бітні числа.

3) Формування об’єднання, яке містить всі унікальні k-мери з обох послідовностей.

4) Ініціалізація структури «ключ-значення» для кожної послідовності.

5) Використання елементів з об’єднання як ключів у створеній структурі.

6) Обхід кожної з послідовностей із додаванням +1 до відповідного значення ключа при знайдені k-меру.

7) Обчислення k-мер відстані.

Алгоритм на основі обчислення мінімальних гешів MinHashDistCount:

Вхід: Дві послідовності (ДНК, ПВП, ВП у відповідних форматах), для яких потрібно оцінити схожість.

Вихід: Обчислена MinHash відстань, що характеризує ступінь схожості.

Послідовність дій:

1) Розбиття послідовностей на k-мери (довжина k визначається за формулою (4.1); для $k<33$ при аналізі ДНК та $k<65$ для двійкових послідовностей).

2) Для менших розмірів k – перетворення у двійкове представлення згідно з правилами (див. розділ 3) із подальшим конвертуванням у 64-бітні числа. Для більших k-мерів – представлення у двійковому вигляді без конвертації.

3) Гешування отриманих значень.

4) Сортування гешів за зростанням.

5) Вибір певної кількості найменших гешів для представлення послідовності.

6) Обчислення MinHash відстані.