НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Звіт за темою

ДОСЛІДЖЕННЯ СХЕМ ГЕНЕРАЦІЇ ПВЧ ДЛЯ ІНТЕЛЕКТУАЛЬНОЇ КАРТКИ, ТОКЕНА ТА СМАРТФОНУ

Виконали студенти
групи ФІ-32мн
Баєвський Констянтин,
Шифрін Денис,
Кріпака Ілля

Київ — 2023

ЗМІСТ

ВСТУП

В епоху неспинно зростаючого розвитку технологій у сучасному світі, питання безпеки даних і персональної інформації користувачів стає невід'ємною частиною розробки сучасних продуктів та екосистем. Еволюція типів з'єднання пристроїв та підвищення рівню різноманітності атак і шкідливого програмного забезпечення роблять захист інформації критично важливим.

В основі безпечних систем лежать випадкові числа, які відіграють ключову роль у криптографічних операціях. Ці числа використовуються для створення надійних ключів шифрування, встановлення початкових значень і лічильників, а також параметрів протоколів. Ступінь випадковості цих чисел безпосередньо впливає на безпеку системи, і будь-яка слабкість у процесі їхньої генерації може призвести до відкриття дверей для атак, які можуть скомпрометувати ключі, перехопити дані та в кінцевому підсумку зламати пристрої та їхні комунікації.

Розробка генераторів істинних випадкових чисел, що забезпечують стабільно високу якість ентропії під час зміни процесів, температури, напруги і частоти, дуже складний процес. Для забезпечення найвищої якості міжнародні органи стандартизації розробили критерії, що дають змогу підтвердити істинно випадковий характер ГВЧ перевіреним і статистично суворим чином.

У роботі розглядається важливість TRNG [1], які використовуються для генерації ПВЧ для інтелектуальних карт [2], токенів та смартфонів, описуються їхні основні компоненти та характеристики. Також наводиться приклад статті із дослідженням генераторів певної структури на основі TRNG [3].

1 ОСНОВНІ ВІДОМОСТІ ПРО TRNG

1.1 Що таке TRNG і чому він важливий?

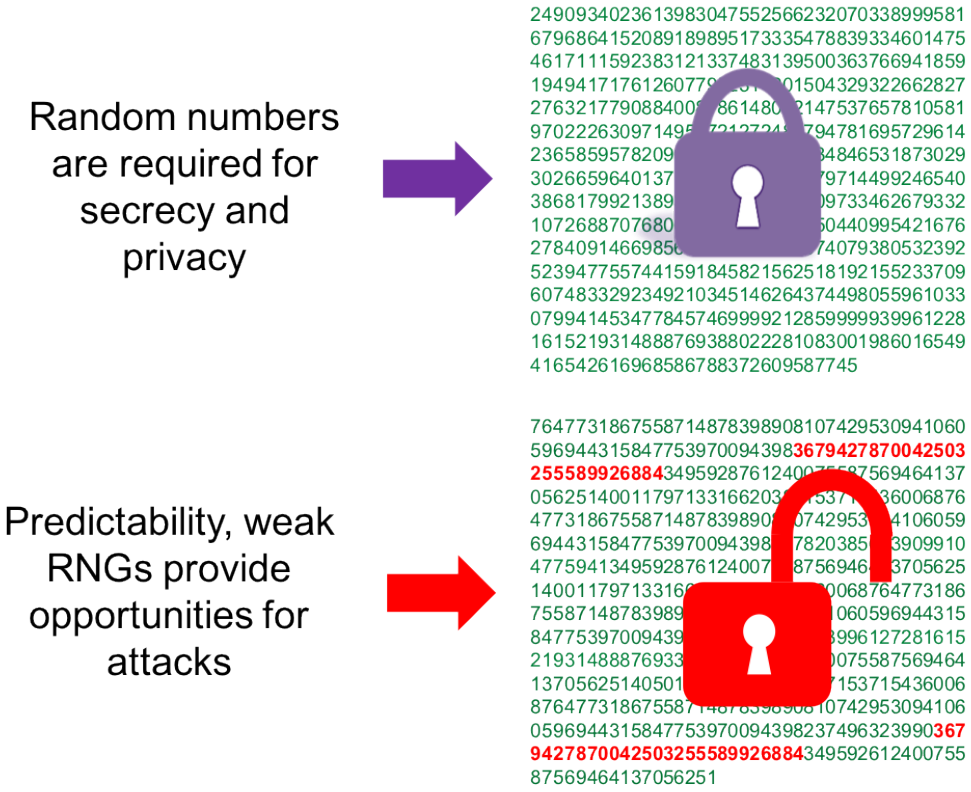
TRNG (True Random Number Generators) - це функція або пристрій, заснований на непередбачуваному фізичному явищі, що назвиний джерелом ентропії, яке призначене для генерування недетермінованих даних (наприклад, послідовності чисел) для запуску алгоритмів безпеки.

Під'єднані пристрої стають частиною повсякденного життя, і ми очікуємо, що вони працюватимуть коректно, захищаючи ділову та особисту інформацію. PRNG лежать в основі захисту цих пристроїв, оскільки вони використовуються для створення і захисту секретів та іншої конфіденційної інформації. Вони є частиною "ланцюжка довіри який необхідно створити, починаючи з SoC, переходячи до рівнів додатків і комунікації з хмарою. Ланцюжок довіри сильний лише настільки, наскільки сильна його найслабша ланка.

Непередбачувані генератори випадкових чисел (ГВЧ) відкривають двері для багатьох можливих атак, які можуть зламати пристрої та скомпрометувати дані. Щоб бути ефективними, випадкові числа мають бути непередбачуваними, статистично незалежними (не пов'язаними з раніше згенерованими випадковими числами), рівномірно розподіленими (однакова ймовірність генерації будь-якого числа) і захищеними

1.2 Де вони можуть бути використані?

Справжні випадкові числа використовуються в таких сферах, як азартні ігри та криптографія, де випадковість має вирішальне значення. Наприклад, багато криптографічних алгоритмів і протоколів безпеки залежать від ключів, а їхня міцність визначається кількістю бітів ключа, які необхідно визначити зловмиснику, щоб зламати систему. Якщо ключі



скомпрометовані, то під загрозою опиняється безпека всієї системи.

Справжні випадкові числа потрібні в різних сценаріях безпеки:

– Генерація ключів для різних алгоритмів (симетричних, асиметричних, MAC) і протоколів (SSL/TLS, SSH, WiFi, LTE, IPsec тощо).

– Виробництво чипів (завдання унікальних ключів пристрою і платформи).

– Початкові значення (для алгоритмів шифрування і MAC, значень TCP-пакетів тощо).

– Генерація nonce і початкові значення лічильників для різних криптографічних функцій.

– Виклики, що використовуються для обміну автентифікацією в протоколі.

– Введення рандомізації для рішень з протидії побічним каналам для захисту від фізичних атак.

1.3 Стандарти для TRNG

Кілька асоціацій зі стандартизації та сертифікації розробляють специфікації та методи перевірки для TRNG, щоб визначити рекомендації щодо розроблення та сертифікації дійсно випадкових рішень.

Американський Національний інститут стандартів і технологій (NIST) розробив набір стандартів NIST SP 800-90A/B/c ("c"поки що перебуває на стадії проєкту) для визначення критеріїв статистичного аналізу, яким має задовольняти TRNG, щоб вважатися достатньо випадковим для криптографічних застосувань. Німецький орган зі стандартизації, Bundesamt für Sicherheit in der Informationstechnik (BSI), уже давно має окремий набір стандартів на TRNG (AIS 20/31).

Обидва стандарти слугують для відсіювання генераторів, що лише здються випадковими, але можуть мати статистичні недоліки, які можуть підірвати безпеку системи. Однак, хоча ці стандарти дають деякі рекомендації щодо архітектури високого рівня, вони не описують конкретно, як створити TRNG; тільки як перевірити, чи працює він. Деталі реалізації залишені на розсуд розробників, що допускає безліч альтернативних підходів. Однак у всіх випадках TRNG повинні відповідати чотирьом критеріям, про які йшлося раніше: вони повинні бути непередбачуваними, одноманітними, незалежними та такими, що їх неможливо виявити.

NIST	National Institute of Standards and Technology (US):
SP 800-90A (Jun 2015)	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
	Defines DRBG/PRNG requirements • Health test requirements • Internal state tracking • Set of “Approved” functions and methods
SP 800-90B (Jan 2018)	Recommendation for the Entropy Sources Used for Random Bit Generation
	Defines NRBG entropy source requirements • Health test requirements • Conditioning recommendations (optional)
SP 800-90c (draft)	Recommendation for Random Bit Generator Constructions
	Defines complete NRBG solution requirements. Combine -90B NRBG with -90A DRBG • Additional health tests and construction methods

BSI	Federal Office for Information Security (Germany/EU):
AIS 20	Functionality Classes and Evaluation Methodology for Deterministic RNGs
AIS 31	Functionality Classes and Evaluation Methodology for Physical RNGs
AIS 20-31 (Sep 2011)	Combined specifications

На додаток до тестів відповідності NIST SP 800-90A/B/c і BIS AIS 20-31, NIST випустив набір статистичних тестів для генераторів випадкових і псевдовипадкових чисел для криптографічних додатків під назвою NIST SP800-20. Однак цих тестів недостатньо для виявлення деяких слабких місць у генераторах випадкових чисел, тому було розроблено інші тести, що доповнюють перевірку рандомізації для TRNG, зокрема набори Diehard і Dieharder.

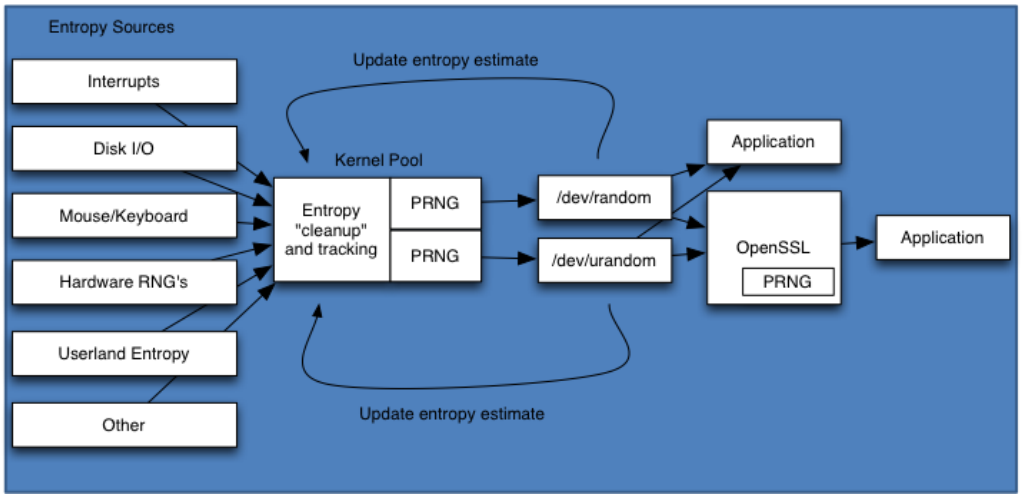
Такі сертифікати, як американський федеральний стандарт обробки інформації (FIPS) 140-2 і нещодавно вийшовший 140-3, Common Criteria (CC) і китайський Office of State Commercial Cryptography Administration (OSCCA), покликані забезпечити повну відповідність кінцевих продуктів вимогам, зазначеним у специфікаціях. Спеціалізовані лабораторії перевіряють архітектуру TRNG, оцінюють властивості генерації випадковості, тестують і підтверджують, що продукти дійсно відповідають вимогам.

1.4 Рішення TRNG

Справжньої випадковості домогтися дуже складно. Правильно сконструйований TRNG повинен збирати ентропію з якого-небудь випадкового процесу (наприклад, шуму, створюваного струмом у транзисторі, або часу між подіями радіоактивного розпаду), а потім кондиціонувати ентропійний сигнал, щоб видалити зміщення і вибілити спектр отриманої послідовності виходів. Цей процес потрібно контролювати з урахуванням таких факторів, як робоча температура, старіння, сприйнятливість до електронних шумів і збоїв, зміна напруги та діапазон робочих частот. Без контролю цих чинників схема TRNG може бути модифікована сторонніми особами, які намагаються вплинути на її роботу.

Одним із прикладів архітектури RNG є запуск генератора псевдовипадкових чисел (PRNG) криптографічної якості з невідомим початковим значенням, а потім використання PRNG протягом певного

періоду часу або для отримання певної кількості випадкових даних. Потім PRNG буде повторно засіяний і знову використаний протягом деякого часу, і так далі. Як затравку для PRNG слід використовувати секретний випадковий вхідний сигнал, отриманий від "джерела ентропії такого як високоякісний TRNG.



Як можна бачити, сенс використання PRNG і TRNG разом, можливо, для створення CSPRNG, полягає саме в цьому: після отримання достатньої початкової випадковості, швидкість, рівномірність і ентропія можуть бути досягнуті шляхом безперервного заповнення пулів навіть досить низькоякісними TRNG, а потім вилученням результатів за допомогою ефективного PRNG, наприклад, легкого симетричного шифру або хеш-функції.

Компанія Synopsys, до слова, розробила відповідні стандартам і готові до сертифікації TRNG, які можна застосувати до будь-яких цифрових напівпровідникових пристроїв і які добре переносяться на будь-які ASIC і більшість FPGA-технологій. TRNG широко впроваджуються аж до 5-нм техпроцесів, конфігуруються замовником і підтримують цілу низку привабливих функцій, включно з широким динамічним діапазоном системного тактового генератора, резервуванням та вибором кількості внутрішніх генераторів сідінгу, автоматичним та ручним перезапуском, вихідними потоками для захисту від побічних каналів та різними

інтерфейсами (з відображенням у пам'яті, послідовним та nonce, який підходить для модулів захисту контенту HDCP 2.3):

Ядро генератора істинних випадкових чисел DesignWare® належить до класу недетермінованих генераторів випадкових бітів (NRBG). Ядро містить джерело ентропії та схему відбілювання, які генерують рівномірно розподілену випадкову послідовність бітів. Вихідні дані DesignWare TRNG можна використовувати безпосередньо або для засіву/заправлення детермінованого генератора випадкових бітів (DRBG), схваленого NIST SP 800-90A, залежно від сфери застосування. Випадкові дані, що генеруються DesignWare TRNG, мають бути статистично еквівалентними рівномірно розподіленому шуму. Схема містить генератор сідінгу, який створює недетерміноване випадкове значення для запуску TRNG.

Генератор випадкових чисел DesignWare True Random Number Generator Core for NIST SP 800-90c повністю відповідає специфікаціям NIST SPA800-90A/B/c і BSI AIS 20/31. Він генерує випадкові числа, які статистично еквівалентні рівномірно розподіленому потоку даних. Ядро містить схему кондиціонування, схвалену NIST SP800-90B, із сумісним джерелом шуму і DRBG, схвалену NIST SP800-90A. Ядро підтримує високопродуктивні операції (3,2 Гбіт/с за частоти 500 МГц) для генерації випадкових чисел, які мають бути статистично еквівалентними рівномірно розподіленому потоку даних. За кремнієвої реалізації DesignWare TRNG може відповідати найвищим комерційним і державним стандартам і підтримувати сертифікацію кінцевих продуктів, включно з FIPS 140-2 / 140-3, Common Criteria і OSCCA. У ядро можна включити до 8 віртуальних TRNG, що забезпечує можливість безпечного доступу до випадкових чисел між декількома користувачами, наприклад, у багатоядерній процесорній системі. IC також підтримує збір фонового шуму для генерації нової ентропії у фоновому режимі та збереження її для наступної операції сідінгу, що усуває час очікування наступного засіву.

2 ГЕНЕРАЦІЯ ВИПАДКОВИХ ЧИСЕЛ У СМАРТ КАРТАХ

Смарт-картки широко використовуються в різних галузях, таких як банківська справа, телекомунікації, посвідчення особи та інші, де потрібна генерація унікальних і безпечних даних для шифрування, підписів та інших криптографічних сервісів. У зв'язку з цим, дослідження генерації псевдовипадкових чисел (ПВЧ) у смарт-картках не таке вже й безпідставне.

Розглянемо, як TRNG реалізовано в комп'ютерах загального призначення, і як їхня архітектура може бути адаптована для використання в смарт-картках. Модифікації моделі генератора TRNG, реалізованих на Java Card, аналізуються з погляду продуктивності та випадковості за використання набору статистичних тестів NIST SP 800-22.

Важливою мотивацією є необхідність забезпечення безпеки генерованих послідовностей, особливо в умовах обмежених ресурсів смарт-карток. Потрібно звертати увагу на критерії безпеки, що висуваються до генераторів PRNG, включно з вимогами Common Criteria та AIS31. Поглянемо на різні аспекти генерації PRNG у смарт-картках, модифікації наявних моделей та їхній аналіз для забезпечення безпеки та ефективності в контексті криптографічних застосувань.

2.1 Пристрій смарт-картки

Смарт-картка являє собою пристрій з обмеженими ресурсами, включно з центральним процесором, пам'яттю тільки для читання (ПЗП), оперативною пам'яттю (ОЗП), програмованою пам'яттю для читання, яка електрично стирається (EEPROM), і криптопроцесором. Дані зберігаються в ПЗП і EEPROM, де останнє являє собою обмежений за циклами ресурс, що підкреслює необхідність ефективного захисту від атак.

Обмеження смарт-карт на генератори псевдовипадкових чисел

пов'язані з обмеженими джерелами ентропії. На відміну від комп'ютерів, де джерела ентропії більш різноманітні, у смарт-картах ці джерела обмежені. Апаратні обмеження також включають обмежений обсяг пам'яті та обмежені обчислювальні можливості. У зв'язку з цим, кращим рішенням є використання апаратних генераторів псевдовипадкових чисел. Існуючі генератори випадкових чисел у смарт-картках зазвичай засновані на регістрах зсуву з лінійним зворотним зв'язком (LFSR) [4], але такі послідовності не забезпечують достатнього рівня безпеки для криптографічного використання. Крім того, обмеження стандарту FIPS 140-2 впливають на вибір програмних реалізацій TRNG у смарт-картках. Важливим моментом є відсутність можливості противника надавати або маніпулювати вхідними даними TRNG, що забезпечує стійкість до передбачень і запобігає розкриттю інформації про внутрішній стан генератора. Такі обмеження смарт-карток вимагають ретельного підходу до розробки та забезпечення безпеки генераторів псевдовипадкових чисел.

2.2 Загальна модель для TRNG

Розглянемо загальну модель для генераторів псевдовипадкових чисел (TRNG).

1) **Узагальнена модель TRNG:**

а) Розроблено загальну модель, враховуючи бажані властивості та рекомендації з різних джерел.

б) Модель включає в себе функцію форматування введення (IFF), алгоритм-генератор (GA), генератор оновлення насіння (SUG), і функцію форматування виходу (OFF).

2) **Кроки обробки:**

а) IFF: Витягує початкове значення з циклічного буфера (CB), що слугує джерелом ентропії. Обробляє дані, задовольняючи вимогам алгоритму-генератора.

б) GA: Приймає вихідні дані IFF і обробляє їх з використанням

криптографічної функції (Hash, MAC, DES, AES та ін.).

в) SUG: Генерує значення для оновлення насіння і повертає його IFF, використовуючи XOR з виходом GA.

г) SFUF: Виконує оновлення насіння, отримуючи значення оновлення і виконуючи XOR з виходом SUG.

д) OFF: Приймає вихідні дані GA і форматує їх відповідно до вимог користувача або програми.

3) **Ітераційний процес:**

а) Другий раунд включає кроки 1 і 2, але вихід GA не передається SUG.

б) Додаткова ітерація (OFF) виконується для приховування значення, яке використовується для оновлення затравочного файлу.

4) **Формування Псевдовипадкових Чисел:**

а) Функція форматування виходу (OFF) перетворює вихідні дані GA в бажаний вихідний сигнал.

2.3 Результати експериментів

Тепер надамо результати роботи шести реалізацій генераторів псевдовипадкових чисел (TRNG) для смарт-карток на основі представленої загальної моделі від авторів статті. Реалізації різняться не тільки використовуваним алгоритмом генератора (GA), а й пов'язаними з ним функціями. Весь експеримент проводився на Java Card, як на симуляторі, так і на реальних картах.

Експериментальне середовище:

– Тести проведено на симуляторі (Java Card simulator JCWDE) і реальних смарт-картках.

– Експерименти підтримують використання різних криптографічних алгоритмів як GA.

Вимірювання продуктивності:

– Продуктивність тестованих TRNG оцінювалася на Java Card Virtual

Machine.

– Експериментальна матриця продуктивності включає результати вимірювань для кожної реалізації на двох 16-бітних Java-картах.

Оцінка розмірів:

– Розміри реалізацій в EEPROM представлені в таблиці. У комерційній реалізації в EEPROM зберігається тільки файл приманки, а код ГПСЧ, частина операційної системи смарт-картки, міститься в ПЗП.

Тести Продуктивності:

– Смарт-картки вставлялися в зчитувач, підключений до осцилографа.

– Тестовий додаток вибирав TRNG і запитував 128 біт випадкового числа.

– Вимірювався час від запиту до отримання вихідного сигналу, а потім бралося середнє значення для оцінки продуктивності.

Результати:

– Генератори на основі хеша показали найкращу продуктивність серед реалізованих TRNG.

– Матриця продуктивності та статистичні тести для випадковості надають повний огляд характеристик кожної реалізації.

ALGORITHMS WITH INPUT REQUIREMENTS AND PRODUCED OUTPUT LENGTH

Generator	Program Size	Seed File Size
SHA - 1	2086 bytes	550 bytes
SHA - 256	2123 bytes	550 bytes
HMAC - DES	2283 bytes	160 bytes
HMAC - AES	2299 bytes	160 bytes
DES	2245 bytes	160 bytes
AES	2189 bytes	160 bytes

TABLE III
PERFORMANCE MEASURE (MILISECONDS)

Performance Measure		SHA-1	SHA-256	HMAC-DES	HMAC-AES	DES	AES
Card One	*Average Time*	43.85	48.39	272.29	273.29	156.56	154.98
	Fastest Time	40.27	44.83	267.80	269.17	151.76	150.00
	Slowest Time	55.57	53.39	287.89	282.80	168.19	165.26
	Standard Deviation	4.48	3.13	5.12	4.22	4.21	4.50
Card Two	*Average Time*	42.84	47.64	263.87	263.02	156.56	148.02
	Fastest Time	39.45	44.08	257.32	257.32	151.76	144.54
	Slowest Time	54.01	58.86	272.17	272.11	168.19	152.01
	Standard Deviation	4.40	4.23	3.57	3.70	4.21	2.37

TABLE IV
NIST STATISTICAL TEST RESULTS (PERCENTAGE OF PASSING SEQUENCES)

Algo / NIST Tests	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SHA-1	98.60	98.99	99.00	98.00	100	99.04	100	100	100	98.00	100	99.00	98.81	99.47
SHA-256	99.30	100	97.00	97.00	98.97	98.78	100	98.00	99.00	99.50	99.00	98.00	98.86	100
HMAC-DES	99.20	100	97.00	99.00	98.97	99.04	99.00	100	98.00	100	100	98.00	100	97.62
HMAC-AES	99.30	100	99.00	100	100	99.01	98.00	97.00	97.00	99.50	94.00	98.50	99.04	100
DES	99.20	98.99	100	100	100	97.98	100	99.00	100	99.00	97.00	99.00	96.67	96.67
AES	99.20	97.98	96.00	97.00	98.97	97.98	98.00	99.00	99.00	100	96.00	98.50	100	100

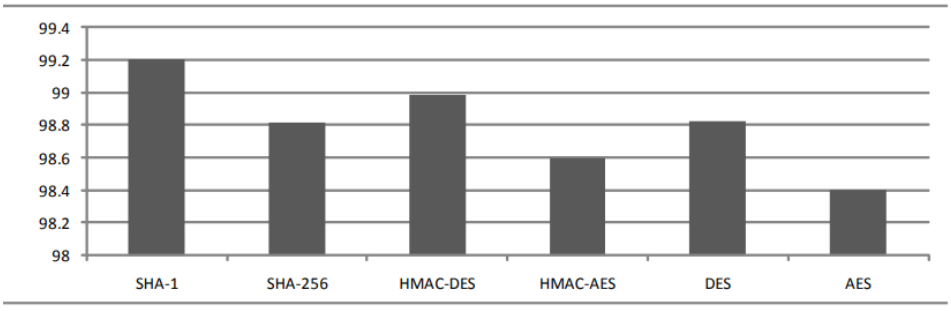


Рисунок 2.1 – Graph Depicting Average Percentage of Passing Sequence Across NIST Statistical Test Results (Table IV)

ВИСНОВКИ

Підключені пристрої та їхні комунікації мають бути захищені від загроз і атак, що розвиваються, для захисту екосистем і цінної особистої та ділової інформації. Високоякісні TRNG є фундаментальною технологією, необхідною для побудови ланцюжка довіри в системах, оскільки багато криптографічних алгоритмів шифрування/автентифікації та протоколів безпеки залежать від істинних випадкових чисел для генерації ключів, викликів і початкових значень. Загальна безпека систем і додатків залежить від якості джерела ентропії, яке забезпечують TRNG. Недоліки генераторів випадкових чисел можуть бути використані зловмисниками для компрометації пристроїв, які в іншому алгоритмічно безпечні. Ефективні ГВЧ повинні відповідати стандартам NIST і AIS і можуть бути сертифіковані в кінцевих продуктах за такими сертифікатами, як FIPS 140-2/140-3, Common Criteria (CC) і китайський OSCCA.

Генерація випадкових чисел це необхідне і складне завдання, для якого пропонується безліч рішень. Однак будь-яка сучасна обчислювальна система (включно з IoT-пристроями), якщо їй потрібні випадкові числа, матиме у своєму складі TRNG. Хоча мінімальна ентропія, яку можна витягти з кожного типу випадкових подій за певний проміжок часу, може сильно відрізнятися для кожного з них. TRNG самі по собі зазвичай не підходять для багатьох криптографічних цілей, оскільки істинність їхньої випадковості не має на увазі незміщену рівномірність їхніх виходів. Ця нерівномірність може бути особливо важливою під час переведення зі сфери вимірювань у сферу необхідної випадковості.

Однак, сучасні вимоги безпеки стикаються з обмеженнями генераторів істинних випадкових чисел ([5], [6], [7]). Використання фізичних джерел у TRNG призводить до високого енергоспоживання й обмеженої пропускної здатності, що стає неприйнятним для сучасних інтегрованих систем. TRNG також чутливі до змін умов експлуатації, що вимагає додаткової постобробки

для забезпечення стабільності вихідних даних.

Але підхід майже завжди полягає у використанні як TRGN для вибірки випадковості, так і PRGN або DRGN для змішування нової випадковості зі збереженою випадковістю в унікальні, рівномірні та статистично незалежні вихідні дані, які не піддаються вгадуванню. У такій системі, яка добре спроектована і реалізована, доступ до будь-якої кількості виходів не повинен допомагати в вгадуванні минулих або майбутніх виходів. Якщо в системі використовується тільки TRGN без будь-якого PRGN, то варто ставитися з великою підозрою до такої системи. Загальноприйнято, що чим швидше, рівномірніше розподілені, ентропійніше й ефективніше, тим краще. Але сенс спільного використання TRGN і PRGN, можливо, для створення CSPRNG, полягає в наступному: Після отримання достатньої кількості вихідної випадковості швидкість, рівномірність і ентропію можна досягти шляхом постійного поповнення пулів навіть досить низькоякісними TRGN, а потім вилучення вихідних даних за допомогою ефективного PRGN, наприклад, легкого симетричного шифру або хеш-функції. Так, наприклад, алгоритм Hash DRBG з використанням SHA-256 може бути обраний для розробки ядра CSPRNG. Цей вибір забезпечує необхідний рівень безпеки за прийнятної пропускної здатності.

ЛІТЕРАТУРА

1. Synopsys. True Random Number Generator Security. *Synopsys Technical Bulletin.* https://www.synopsys.com/designware-ip/technical-bulletin/true-random-number-generator-security-2019q3.html (2019).
2. Pannetrat, A. & Cunche, M. *True Random Number Generator for Smart Cards* в (2015). https://www.researchgate.net/publication/275727841_True_Random_Number_Generator_for_Smart_Cards.
3. He, D., Luo, X., Chen, G. & Chen, C. Pseudorandom Number Generation in Smart Cards: An Implementation Performance and Randomness Analysis. *Journal of Signal Processing Systems* **70.** https://www.researchgate.net/publication/235344028_Pseudorandom_Number_Generation_in_Smart_Cards_An_Implementation_Performance_and_Randomness_Analysis, 197—208 (2013).
4. Kar, R., Sur-Kolay, S. & Maitra, S. General Models for Pseudorandom Number Generators in Smart Cards. *Journal of Hardware and Systems Security* **3,** 73—96. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7180860/ (2019).
5. User. *Are TRNGs used in low-power devices? Why?* Crypto Stack Exchange Question. https://crypto.stackexchange.com/questions/107713/are-trngs-used-in-low-power-devices-why.
6. Goubin, L. & Patarin, J. *True Random Number Generators in Smart Cards* в (2000). https://link.springer.com/chapter/10.1007/3-540-44598-6_20.
7. Chmielewski, L. & Kryszkiewicz, M. *A True Random Number Generator for Smart Cards* в (2007). https://ieeexplore.ieee.org/document/4221841.