DOI: 10.18372/2310-5461.47.14934

УДК 519.212.2:681.51: 621.317

С. В. Поперешняк, канд. фіз.-мат. наук, доц.
Київський національний університет імені Тараса Шевченка
orcid.org/0000-0002-0531-9809
e-mail: spopereshnyak@gmail.com

ТЕСТУВАННЯ ДАТЧИКІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ, ВБУДОВАНИХ В СМАРТ-КАРТИ

Вступ

Сучасні інформаційні технології передачі даних здебільшого були побудовані з обов'язковим використанням особистого ідентифікатора, в якості якого найбільш доречно використовувати електроні пластикові картки, або смарт-карти, у різних форматах.

Особливістю електронних пластикових карток є наявність в їх складі генераторів випадкових чисел, на основі яких формуються ключі шифрування. Якість роботи генераторів випадкових чисел визначає якість ключів, а загалом крипостійкість даних, що передаються.

Випадкові числа та їх генератори є невід'ємними елементами сучасних смарт-карт для банківських систем. У смарт-картах через свою конструкцію можлива як програмна генерація псевдовипадкових послідовностей, так і апаратна. Датчики псевдовипадкових послідовностей відіграють важливу роль у генерації різних типів випадкових даних (наприклад, ключі, початкові значення для генерації послідовностей тощо) [1].

Тому викликає інтерес вибір і формування системи тестів за допомогою яких доцільно оцінювати якість роботи генераторів псевдовипадкових чисел смарт-карт різних виробників.

У даній статті проводиться дослідження датчиків випадкових чисел, які застосовуються в захисті інформації та системах контролю доступу. Для тестування були взяті такі статистичні пакети і окремі критерії:

а) статистичний пакет NIST;
б) статистичні тести Д. Кнута;
в) статистичні тести Crypt-X ;
г) статистичний пакет U01;
д) статистичний пакет DIEHARD;
є) багатовимірні статистики.

Постановка проблеми

В останнє десятиліття, окрім використання технології смарт-карт у банківській та телекомунікаційній галузях, така технологія отримала прийняття в інших галузях: електроні ключі, посвідчення водія, національні посвідчення особи, посвідчення особи студента, соціальне забезпечення, здоров'я, проїзні документи та паспорти. У більшості цих полів смарт-карти використовують програми, які потребують генерації сеансових ключів, підписів DSS та псевдовипадкових послідовностей бітів тощо. Ці функції потребують безпечного та непередбачуваного генератора псевдовипадкових чисел для їх безпеки [2].

Стаття вивчає випадковість і як комп'ютери загального призначення генерують її. В статті також розглядаються вимоги безпеки, яким повинен задовольняти генератор псевдовипадкових чисел (ГПВЧ) для використання в криптографічних програмах. А також особлива увага приділяється відомим та новим методам тестування випадкових бітових послідовностей.

Для виявлення закономірностей аналізованих ПВП (або до їх відрізків різної довжини) застосовують широкий спектр різних статистичних тестів, розроблених в останні десятиліття. Попередній аналіз розглянутих статистичних тестів дозволив визначити, які з них найбільш придатні для використання в різних завданнях розробки засобів криптографічного захисту інформації.

Аналіз існуючих пакетів для статистичного тестування призводить до висновку, що багато тестів можуть успішно використовуватися для дослідження ПВП на випадковість. Разом з тим, особливості прикладних задач показують, що класична математична модель статистичного тестування не цілком адекватно відображає потреби в дослідженні деяких об'єктів на випадковість. Така ситуація виникає, коли генеруються послідовності невеликої довжини (до 100 біт). У такому разі досліджувану ПВП неможливо коректно протестувати за допомогою одновимірної статистики. У цій ситуації пропонуємо протестувати ланцюжок на випадковість використовуючи дво- і/або тривимірні статистики.

Аналіз останніх досліджень і публікацій

Більшість криптографічних механізмів потребує певних даних від генератора випадкових чисел. Але слабкість генератора з часом може

послабити увесь криптографічний механізм. Якщо генератор слабкий, це може призвести до незахищеності всього механізму [3]. Це видно з результатів, показаних у працях [4–7].

Щоб генератор був криптографічно захищеним, генеровані послідовності повинні бути випадковими, так щоб противник був не в змозі передбачити майбутнє або визначити минулі послідовності.

Кращим вибором є встановлення на основі апаратних даних істинного генератора випадкових чисел (ІГВЧ) [8–10]. Однак на пристрої з обмеженими ресурсами, як-от смарт-карти, ІГВЧ може бути вразливим до аналізу невідповідності [11; 12]. Крім того, ряд стандартів мають суворі вимоги до ІВБГ. Наприклад, у праці [13] (німецька схема) вимоги до ІГВЧ визначені в AIS31 [14]. У ньому йдеться про те, що при кожному включенні генератора слід проводити випробування, визначені у FIPS 140-2 [15]. Це накладає величезні обчислювальні витрати на механізм ІГВЧ, а також може негативно вплинути на безпеку генератора від атак, таких як SPA/DPA [16] та шаблонні атаки [17].

Як обговорювалося в праці [18], ці жорсткі вимоги ускладнюють ефективний та корисний ІГВЧ. Тому альтернативою було б мати генератор псевдовипадкових чисел (ГПВЧ). Реалізація ГПВЧ в комп'ютерах зазвичай тестується за допомогою ряду статистичних тестів (наприклад, у наборі тестів NIST SP 800-22 є 15 тестів). Однак загальні критерії (німецька схема) детально описує перелік випробувань в AIS20 [19], які складаються лише з п'яти тестів. Причиною цього можуть бути обмежені ресурси смарт-карти. Однак мати впевненість у тому, що ГПВЧ на смарт-картах однаково виробляє випадкові послідовності, як ГПВЧ на комп'ютері.

ГПВЧ на основі інтелектуальної карти також слід ретельно перевіряти, використовуючи ті самі статистичні тестові тести, що і ГПВЧ на комп'ютері, але врахувавши обмеження на довжину бітової послідовності.

У цій статті пропонуємо тестування ГПВЧ здійснювати за допомогою багатовимірних статистик, використовуючи певні бітові шаблони. Крім того, всі ці ГПВЧ тестуються за допомогою тестового набору NIST SP 800-22.

Аналіз останніх досліджень і публікацій показує широкий інтерес до безпеки в смарт-картах і, як наслідок, в останній час, запропоновано різні підходи до побудови полегшеного генератору псевдовипадкових чисел, який може застосовуватися в пристроях з певними обмеженими характеристиками.

Як показує аналіз, якість даних ГПВЧ перевіряються за допомогою набору статистичних тестів, які були розроблені в останні десятиліття і набули досить широкої популярності, а також з використанням багатовимірних статистик.

Мета статті. Розглянути схему функціонування ГПВЧ в обмежених пристроях. Виділити основні вимоги до сучасних смарт-карт.

Розглянути сумісні розподіли 2-ланцюжків і 3-ланцюжків фіксованого виду випадкової бітової послідовності, які дозволяють проводити статистичний аналіз локальних ділянок цієї послідовності.

Метою роботи є розгляд існуючих систем статистичних тестів і формування системи тестів з використанням багатовимірних статистик для оцінки якості роботи генераторів псевдовипадкових чисел, що використовуються в обмежених пристроях, наприклад, у смарт-картах.

Наукова новизна

У статті запропоновано критерій для перевірки на випадковість бітових послідовностей невеликої довжини (до 100 біт). Даний підхід доцільно використовувати для тестування полегшеного генератора псевдовипадкових чисел в пристроях з певними обмеженнями на ресурси.

Псевдовипадкові числа

У цьому розділі ми спочатку пояснюємо псевдовипадкові числа, а потім їх генерування.

Генератор псевдовипадкових чисел

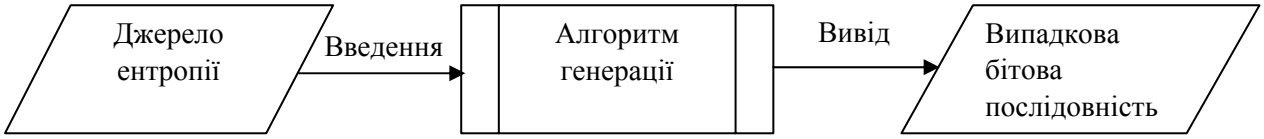
Генератор псевдовипадкових чисел ГПВЧ, визначений у праці [2], є «алгоритмом, який задає справді випадкову бінарну послідовність довжини k, виводить двійкову послідовність довжиною $l >> k$, яка видається випадковою».

Згідно з цим визначенням, на елементарному рівні генератор складається з функції, що називається *алгоритмом генерації*, яка приймає невеликий випадковий рядок і перетворює його в більш довгий рядок, який статистично незалежний від вхідної рядка.

Вхід до алгоритму генерації відомо як зерно, і він має вирішальне значення для випадковості виходу генератора. Значення зерна взяті з джерела ентропії, де ентропія є мірою хаотичності, випадковості або мінливості в закритій системі.

Випадковість або хаотичність відносно спостерігача, якщо спостерігач не в змозі передбачити масштаб розкиду, то джерело ентропії забезпечує високу ентропію [2].

На рисунку показана модель елементарного рівня ГПВЧ.



Узагальнена модель для ГПВЧ

Крім того, ГПВЧ також повинен мати однобічну властивість, що робить обчислювально складним пошук рядка введення зі створеної вихідної послідовності. Нарешті, загальна модель ГПВЧ повинна мати бажані властивості, які пояснюються в наступному розділі. Існують чітко визначені стандарти, такі як NIST SP 800-90 [20; 21] та ISO / IEC 18031 [22], який дає керівництво щодо того, як ГПВЧ може бути реалізований у комп'ютерному середовищі.

Властивості генератора псевдовипадкових чисел

Є п'ять властивостей, які бажано, щоб були присутні у будь-якій реалізації ГПВЧ [2]. Перша властивість функціональна, а інші чотири — властивості безпеки.

1) Псевдовипадкові послідовності, що отримуються на виході ГПВЧ повинні бути обчислювально відмінні від генератора істинно випадкових чисел.

2) Неможливість передбачити попередні значення: навіть при доступі до внутрішнього стану супротивник не повинен мати змогу генерувати минулий результат. Внутрішній стан ГПВЧ — це фактично його пам'ять. Значення, що зберігаються в ньому, можуть бути лічильником, даними, що введені користувачем, заздалегідь визначеними системними значеннями та криптографічними ключами тощо. Усі ці значення використовуються під час процесу генерації, і вони слугують входом до алгоритму генерації разом із початковим кодом.

3) Неможливість передбачити наступне значення: алгоритм забезпечує стійкість передбачення, якщо противник не в змозі передбачити майбутні виходи, навіть після доступу до внутрішнього стану генератора. Для цього генератору слід регулярно проводити перезавантаження. Повторне дослідження означає зміну джерел ентропії при виявленні доступу або в кінці терміну його експлуатації.

4) Затримка в шифруванні: противник не в змозі передбачити минулі результати, навіть якщо він/вона спостерігає всі майбутні результати з певного моменту часу. Його знання про створені послідовності обмежується першим спостереженням, і всі попередні результати завжди залишатимуться прихованими.

5) Таємниця вперед: противник все ще не зможе передбачити майбутні результати навіть після спостереження минулих результатів. Імовірність прогнозування майбутніх значень не повинна збільшуватися в часі порівняно з генерацією наступних значень ГПВЧ. Статистичні тести перевіряють першу бажану властивість, вимірюючи рівень випадковості у створеному виході. Доступно кілька статистичних наборів тестів, таких як Diehard, TestU01, NIST SP 800-22 тощо, які дають лише впевненість у відсутності повторних шаблонів у генерованій послідовності бітів. Однак жоден набір статистичних тестів не може абсолютно підтвердити як потрібен генератор для використання в певній програмі, тобто статистичне тестування не може слугувати заміною криптоаналізу [2].

Технологія смарт-карт

Розглянемо технологію смарт-карт та її обмеження для проектування ГПВЧ.

Обладнання смарт-карт

Смарт-карта — дефіцитна платформа, яка, як правило, складається з центрального процесорного блоку (процесора), лише пам'яті для читання (ROM), оперативної пам'яті з випадковим доступом (оперативної пам'яті), електричної програмованої пам'яті лише для читання, що може стиратися (EEPROM) та криптосопроцесору [23]. На смарт-картці дані можуть зберігатися в ROM та EEPROM. ROM містить операційну систему карти, а EEPROM зберігає дані, які необхідно оновлювати протягом терміну експлуатації карти. Пам'ять EEPROM має обмежені цикли запису/стирання, і вона становить від 100 000 до 1 000 000 циклів протягом усього діапазону робочих температур і напруг [2], після чого вона стає непрацездатною. Протягом усього свого життя смарт-карта залишається у володінні власника картки [24]. Це дозволяє противнику здійснювати атаки на апаратне забезпечення або програмне забезпечення без будь-якого фізичного обмеження доступу. Оскільки основні дані зберігаються в EEPROM, якщо механізму захисту не існує, супротивник може ефективно змінювати значення, збережені в EEPROM, на свій вибір. Однак карти, які надані добросовісним виробником, запропонують відповідні заходи безпеки для запобігання подібних атак.

Обмеження платформи смарт-карт для ГПВЧ

Для будь-якого ГПВЧ необхідне випадкове джерело ентропії, з якого алгоритм генерації приймає початкове значення та генерує псевдовипадкове число. Таким джерелом ентропії на ПК може бути швидкість, шум, записаний мікрофоном, процеси в черзі та їх час/стан. Однак у смарт-картці джерела ентропії обмежені, і карта не може покладатися ні на внутрішній механізм генерації, ні на зовнішнє джерело для забезпечення ентропії, необхідної для ГПВЧ. Окрім проблеми хорошого джерела ентропії, смарт-карти також обмежені обсягом пам'яті та можливостями обробки. Тому ідеальним рішенням буде ГПВЧ на основі апаратних засобів. Поточні генератори випадкових чисел у мікроконтролерах смарт-карт, як правило, базуються на регістрах зрушень лінійного зворотного зв'язку (LFSR), керованих осциляторами з керованою напругою [2]. Однак послідовності, що створюються виключно LFSR, мають високий рівень лінійності, що дозволяє противнику легко здогадуватися про внутрішні значення та передбачати майбутні результати. Тому для криптографічного використання вони не забезпечують необхідного рівня безпеки. Окрім апаратних моделей, ГПВЧ також може базуватися лише на використанні програмного забезпечення. Однак вибір їх обмежений у смарт-картах через:

1) обмежену ємність пам'яті та швидкість обробки смарт-карти.

2) недоступність хороших джерел ентропії

3) недоречності здійснювати повторне генерування з міркувань безпеки.

4) обмеження FIPS 140-2 [15]. Технічно можливе повторне перезавантаження смарт-карти, і мотивація до цього полягає в наданні стійкості передбачення ГПВЧ. Це можна реалізувати, подавши запит на повторне завантаження, коли смарт-карта вставлена в зчитувач карт. Однак якщо положення про зміну даних в генераторі надаються після виготовлення карти, то противник може атакувати цей механізм на свою користь, дозволяючи йому/їй передбачити майбутні значення. Як наслідок, заборона противнику надати або маніпулювати входом до ГПВЧ перешкоджає пошуку внутрішнього стану генератора.

Загальна модель генератора псевдовипадкових чисел на смарт-картах

З урахуванням бажаних властивостей ГПВЧ, що були зазначені вище, приймаючи, а в деяких випадках і змінюючи технічні рекомендації [2; 20] відповідно до середовища смарт-карт, розглянемо загальну модель ГПВЧ.

Обрана архітектура смарт-карти ГПВЧ — це незначна модифікація [15; 23; 24]. Дана модель з мінімальною модифікацією, може бути реалізовано на більшості доступних у продажу смарт-карт.

Загальна модель для генератора псевдовипадкових чисел на смарт-картах [2].

1) Функція форматування вводу отримує значення насіння з Циклічного буфера. Циклічний буфер надає джерело ентропії ГПВЧ у смарт-картах. Зазвичай Циклічний буфер (також може називатися стартовим файлом) зберігає десять випадкових значень насіння, і кожне значення використовується/оновлюється послідовно. Функція форматування вводу обробляє дані для задоволення вхідних вимог алгоритму генератора. Функція форматування вводу веде облік того, яке значення потрібно отримати при кожному виконанні, і це значення називається початковим індексом.

2) Вихід, що генерується Функцією форматування вводу, обробляється Алгоритмом генератора. Алгоритм генератора – це фактично криптографічна функція. Вибір Алгоритму генератора здійснюється разом із розробниками смарт-карт (або реалізаторами).

3) Вихід Алгоритму генератора подається на Генератор оновлення насіння. Генератор оновлення насіння генерує значення, яке використовується в процесі оновлення насіння, а також повернеться до Функції форматування вводу. Основна архітектура Генератор оновлення насіння полягає в тому, що функція «Xor» є входом до Алгоритму генератора з його результатом. Якщо вхід/вихід невідповідний по довжині, то він з'єднує коротшу довжину з собою, поки вона не дорівнює іншій.

4) Вихід генератор оновлення насіння обробляється Функцією оновлення файлів насіння. Функція оновлення файлів насіння отримує значення оновлення (для введення насіння) та «Xor» з виведенням Генератора оновлення насіння. Результат записується в ту саму локацію, з якої функція оновлення файлів насіння отримує значення. Вихід Генератора оновлення насіння повертається до Функції форматування вводу для другого раунду. У другому раунді кроки 1 і 2 виконуються знову. Однак вихід Алгоритму генератора не надсилається в Генератор оновлення насіння, він переходить у Функцію форматування виводу. Ця додаткова ітерація кроків 1 і 2 виконується для приховування значення, яке використовується для оновлення файлу (Циклічний буфер).

5) Функція форматування виводу отримує вхід від Алгоритму генератора і формує цей ви-

хід відповідно до вимог, що сформовані користувачем або додатком. Після того, як він завершить форматувати вхід до потрібного виводу, він надсилає псевдовипадкове число об'єкту.

Огляд інструментів для статистичного тестування ПВП

Для виявлення закономірностей аналізованих ПВП (або до їх відрізків різної довжини) застосовують широкий спектр різних статистичних тестів, розроблених в останні десятиліття. Наведемо відомі набори статистичних тестів.

Серед них найбільш поширені тести Д. Кнута, DIEHARD, CRYPT-X, NIST STS та ін.

Тести Д. Кнута

Одним з перших наборів статистичних тестів був запропонований Д. Кнутом у 1969 році в його класичній роботі «Мистецтво програмування для ЕОМ» [25]. Тести засновані на статистичному критерії. Обчислюване значення статистики порівнюється з табличними результатами, і залежно від імовірності появи такої статистики робиться висновок про її якість. Серед переваг цих тестів — невелика їх кількість і існування швидких алгоритмів виконання. Недолік — невизначеність у трактуванні результатів.

Тести Diehard

Diehard це набір статистичних тестів для вимірювання якості набору випадкових чисел, який розглядають як один з найбільш суворих існуючих наборів тестів. Батарея статистичних тестів призначена для виміру якості ГПЧ, що була створена Джорджом Марсаглія у 1995 році. В основі більшості тестів лежить використання генератора для побудови послідовності відповідно да наданої специфікації і порівняння її характеристик з очікуваними від випадкової. Деякі з наведених випробувань можна виділити в групи за подібністю, а інші являють собою один тест. Більше інформації про тести можна знайти в [26]. Слід відзначити і недоліки даного програмного продукту: немає детального опису тестів і методики трактування їх результатів, більшість тестів є евристичними, проходження тесту має тільки два значення «так» або «ні». Тести Diehard формують на виході числа *p*-значення, які рівномірно розподілені в інтервалі [0; 1], якщо вхідний потік чисел дійсно випадковий. Існує 13 тестів в пакеті Diehard, деякі з них - повторюються за різними параметрами і вони видають 181 *p*-значень загалом.

Тести Crypt-X

Набір статистичних тестів Crypt-X, розроблений дослідниками з науково-дослідного центру з інформаційної безпеки в технологічному університеті Квінсленда в Австралії і є комерційним пакетом програмного забезпечення. Тести застосовуються залежно від типу алгоритму генератора, відповідно спрямовані на тестування генераторів псевдовипадкових чисел. Підтримуються потокові шифри, блокові шифри і генератори потоку ключів. У набір включені такі тести: частотний, на послідовність однакових бітів, лінійна складність, складність послідовності, двійкова похідна, зміна точки.

TestU01

Об'ємна бібліотека тестів на мові С, що включає реалізацію ГПЧ, тести та батареї тестів. Всі випробування що надаються, поділені в групи відповідно до модулів програми.

Один з модулів містить набір статистичних тестів, у який включені класичні тести, інші тести, що містяться в літературі та декілька оригінальних тестів. Другий модуль бібліотеки містить шість визначених підбірок тестів. Три підбірки призначені для тестування послідовності дійсних випадкових чисел, що рівномірно розподілені на проміжку [0, 1]. Три підбірки призначені для бітових послідовностей. Автори підбірок не стверджують, що підбірки містять незалежні тести, які відслідковують усі можливі відхилення від випадковості. Вони вважають, що важко перевірити незалежність тестів і порівняти їх ефективність.

Стандарти і тести NIST

Специфікація та відповідна бібліотека на мові С, що були випущені Інститутом Стандартів та Технологій США [20–21; 27]. Пакет складається з 15 тестів для аналізу бітових послідовностей, що були згенеровані ГПЧ. Відмінність цих тестів від інших сучасних — відкритість алгоритмів. Також серед переваг — однозначна інтерпретація результатів тестування. Вимоги і методика стандарту більше носять технологічний характер. Вони спрямовані на вирішення завдання статистичного контролю псевдовипадкових послідовностей, що використовуються в криптографічних модулях, і в загальному випадку малопридатні до вирішення завдання дослідження статистичних властивостей генераторів.

Проблеми методів

Розглянуті пакети статистичних тестів мають весоме математичне підґрунтя і готову програмну реалізацію. Будь-який з них можна використати для оцінки послідовності або генератора і мати високий рівень впевненості в якості результатів. Однак, в екосистемі статистичних тестів на випадковість можна виділити такі тренди:

- Наявна велика кількість різних тестів та пакетів, що часто підходять до вирішення задачі з зовсім різних сторін;

- Відсутні чіткі лідери, тобто, тести які можна рекомендувати для вирішення більшості проблем;
- Неможливо отримати точний висновок про випадковість послідовності навіть після виконання всіх можливих тестів;
- Майже всі окремі пакети та тести мають деякі обмеження або недоліки.

Логічним буде висновок, що область перевірки випадковості далеко не є завершеною і потребує додаткового дослідження та покращення існуючих підходів. До проблем більшості тестів можна віднести:

– випробування потребують послідовності великої довжини;

– деякі з параметрів тестів неможливо змінити;

– рішення про проходження тесту приймає тільки два значення (так/ні);

– відсутність програмних пакетів для тестування.

Отже, в методах перевірки бітових послідовностей є достатньо проблем для вирішення та підходів для покращення. Особливий інтерес для дослідження складає відсутність тестів, що можуть дати адекватні результати на коротких послідовностях.

Сумісні розподіли числа 2-ланцюжків і числа 3-ланцюжків фіксованого виду

Тести багатовимірних статистик відрізняються тільки шаблонами, на які перевіряється послідовність.

Кожен метод отримує на вхід випадкову бітову послідовність.

Для даної послідовності визначається кількість специфічних шаблонів k_1, k_2 та k_3 (якщо це визначено методом) і виконується обчислення за допомогою формули специфічної для методу.

Розглянемо послідовність випадкових подій

$$\gamma_1 \gamma_2 \dots \gamma_n , \qquad (1)$$

де $\gamma_i = \{0,1\}$, $i = 1, 2, \dots, n$, $n > 0$.

Підпослідовність $\gamma_j \gamma_{j+1} \dots \gamma_{j+s-1}$ послідовності (1) будемо називати s-ланцюжком, $s = 1,2, \dots, n$, $j = 1,2, \dots, n-s+1$.

Позначимо $\eta(t_1 t_2 \dots t_s)$ число s-ланцюжків в послідовності (1), які співпадають з $t_1 t_2 \dots t_s$, де $t_i = \{0,1\}$, $i = 1,2, \dots, s$.

Сформулюємо умову (У).

Умова (У): Послідовність (1) складається з n, $n > 0$, незалежних однаково розподілених випадкових величин; імовірності подій $\{\gamma_i = 1\}$, $\{\gamma_i = 0\}$ відомі та дорівнюють $P\{\gamma_i = 1\} = p$, $P\{\gamma_i = 0\} = q$, $p + q = 1$, $i = 1,2, \dots, n$.

Теорема 1. Нехай виконуються умови (У), n, k_1, k_2, k_3, t, t_1 — цілі числа такі, що $k_1 \ge 0$, $k_2 \ge 0$, $k_3 \ge 0$, $n \ge \max(2k_1, 3)$, $t, t_1 \in \{0,1\}$.

Тоді

$$P\{\eta(tt^*) = k_1, \eta(t1t^*) + \eta(t0t^*) = k_2\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} \sum \prod_{i=0}^{1} C_{k_1}^{\delta_1} C_{m_i-k_1}^{k_1-\delta_i}, \qquad (2)$$

де $m_0 = n - m_1$, символ $\sum$ позначає сумування по всім цілим невід'ємним числам δ_0 і δ_1 таким, що $\delta_0 + \delta_1 = 2k_1 - k_2$, $t^* = 1 - t$;

$$P\{\eta(tt^*) = k_1, \eta(ttt^*) = k_2\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{m_1-k_1}^{k_2} C_{m_0}^{k_1}; \qquad (3)$$

$$P\{\eta(tt^*) = k_1, \eta(tt^*t^*) = k_2\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{m_0-k_1}^{k_2} C_{m_1}^{k_1}; \qquad 4)$$

$$\mathrm{P}\{\eta(tt^*) = k_1, \eta(t1t^*) = k_2, \eta(t0t^*) = k_3\} = \sum_{m_1=k_1}^{n-k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{k_1}^{k_3} C_{m_1-k_1}^{k_2} C_{m_0-k_1}^{k_3}. \qquad (5)$$

Приклад. Для (0,1)-послідовності виду

0 1 1 0 0 1 1 0 1 1 1 0 0 0 0 1 0 1 0 1

значеннями випадкових величин $\eta(tt^*)$, $\eta(t1t^*)$, $\eta(t0t^*)$ і $\eta(t\alpha t^*)$, де $\alpha \in \{0,1\}$, для $t = 1$ є відповідно числа 5, 3, 2 і 5.

Теорема 2. Нехай виконуються умови (У), n, k_1, k_2, k_3, t, t_1 – цілі числа такі, що $k_1 \ge 0$, $k_2 \ge 0$, $k_3 \ge 0$, $n \ge 0$, $t, t_1 \in \{0,1\}$.

Тоді

$$P\left\{\eta(tt^*)=k_1,\eta(t1t)+\eta(t0t)=k_2\right\}=\sum_{m_1=k_1}^{n-k_1}p^{m_1}q^{m_0}\sum_{i\in\{k_1,k_1+1\}}\sum C_{i-1}^{\delta_0}C_i^{\delta_1-m_1+2i}C_{m_0-i+1}^{k_1-\delta_0}\times Z\left(m_1-i;\,m_1-i-\delta_1\right),\quad(6)$$

де $m_0=n-m_1$, символ $\sum$ позначає сумування по всім цілим невід'ємним числам δ_0 і δ_1 таким, що $\delta_0+\delta_1=k_2$, $t^*=1-t$; де

$$Z(a,b)=\begin{Bmatrix}C_{a-1}^{b-1},\text{ якщо }\ a\ge b\ge 0;\\ 1,\text{ якщо }\ a=b=0;\\ 0,\text{ в іншому випадку}\end{Bmatrix}$$

$$P\left\{\eta(tt^*)=k_1,\eta(ttt)=k_2\right\}=\sum_{m_1=k_1}^{n-k_1}p^{m_1}q^{m_0}C_{m_0}^{k_1}\sum_{i\in\{k_1,k_1+1\}}C_i^{m_1-k_2-i}Z\left(m_1-i,m_1-i-k_i\right);\quad(7)$$

$$P\left\{\eta(tt^*)=k_1,\eta(tt^*t)=k_2\right\}=\sum_{m_1=k_1}^{n-k_1}p^{m_1}q^{m_0}\sum_{i\in\{k_1,k_1+1\}}C_i^{k_2}C_{m_0-i}^{k_1-k_2}Z\left(m_1,i+1\right);\quad(8)$$

$$P\left\{\eta(tt^*)=k_1,\eta(ttt)=k_2,\eta(tt^*t)=k_3\right\}=\sum_{m_1=k_1}^{n-k_1}p^{m_1}q^{m_0}\sum_{i\in\{k_1,k_1+1\}}C_i^{k_2-m_1+2i}C_{i-1}^{k_3}C_{m_0-i+1}^{k_1-k_3}Z\left(m_1-i,m_1-i-k_2\right).\quad(9)$$

Результат порівняння тестування ПВП невеликої довжини з використанням статистичного пакету NIST та багатовимірних статистик

Переважна більшість статистичних тестів справді порівнюють декілька вибірок на одній змінній або порівнюють кілька змінних в одній сукупності.

Тепер, якщо додати до цього порівняння кількість змінних чи сукупності та вимірювання декількох змінних, то вибір відповідного статистичного тесту раптом стає надзвичайно простим.

Розглянемо відомі приклади, які наведено у працях [20; 21], а також підхід з використанням багатовимірних статистик [28–31]. І обчислимо ймовірності відповідних подій використовуючи розподіли 2-ланцюжків і 3-ланцюжків фіксованого виду випадкової (0, 1)-послідовності. Покладемо $t=0$. Таким чином розглянемо такі випадкові величини: $\eta(01)$, $\eta(011)$, $\eta(001)$ і $\eta(000)$.

Проаналізуємо таблицю, де використані такі позначення за умові, що $t=0$:

- P, отримана за допомогою відповідного тесту статистичного пакету NIST;
- P_1 - імовірність $P\left\{\eta(tt^*)=k_1,\eta(t1t^*)+\eta(t0t^*)=k_2\right\}$, яку можна обчислити використовуючи співвідношення (2);
- P_2 — імовірність $P\left\{\eta(tt^*)=k_1,\eta(ttt)=k_2\right\}$, яку можна обчислити використовуючи співвідношення (7).

Таблиця

Порівняльна таблиця ймовірностей з використанням тестового пакету NIST

Тест Nist	Вхідні параметри	*P*-value	P_1	P_2
Frequency	1101010011	0,52708	0,14648	0,21191
Block frequency	1011100110 M = 2	0,84914	0,13671	0,18261
Runs	0011001111	0,59816	0,02050	0,18261
Longest run of ones	1101110111	0,00255	0,20507	0,18261
Binary matrix rank	11011101111101010011 M = 2	0,27012	0,09252	0,06847
Spectral	1011010101	0,46815	0,02343	0,04980
Non overlapping template	110101010111101100011 M = 10 T = 111	0,34415	0,05948	0,06863
Overlapping template	10010101101110010001 M = 10 T = 001	0,50979	0,06344	0,06055
Linear complexity	0011101101	0,99482	0,09765	0,21191
Maurers	00010011101001100101 M = 2 L = 4	0,71	0,03172	0,06055

Закінчення таблиці

Тест Nist	Вхідні параметри	*P*-value	P_1	P_2
Entropy	0100100101 M = 3	0,40306	0,02734	0,04980
Serial	1101010011 M = 3	0,80879	0,14648	0,21191
Cumulative sums	1001001010	0,94173	0,14648	0,21191
Random excursions	1101010011	0,89307	0,11718	0,18554
Random excursion variant	0011100001	0,68309	0,13671	0,08691

Як можна побачити з таблиці використання двовимірних або тривимірних статистик дає більше точний результат для подальшого аналізу двійкової послідовності невеликої довжини (до 100 біт). Також зауважимо, що відповідно до обмежень на довжину послідовності, які наведено у праці [20], рекомендована довжина послідовності *n* повинна бути, в більшості випадків, більша, ніж 10^6 біт

Висновки

Потреба перевірки послідовності за допомогою декількох критеріїв неодноразово відзначалася в літературі. Чим більше використовуються комп'ютери, тим більше необхідно випадкових чисел, і генератори випадкових чисел, які раніше вважалися загалом задовільними, тепер недостатньо гарні для застосування у фізиці, комбінаторики, стохастичній геометрії, не кажучи вже про криптографічні додатки. У зв'язку з цим вводяться суворіші критерії, які перевершують класичні методи.

Аналіз ефективності генераторів псевдовипадкових послідовностей є нагальною проблемою смарт-карт за умов використання більш досконалих методів шифрування та захисту інформації. Наявні способи показують низьку гнучкість та універсальність у засобах знаходження прихованих шаблонів у даних. Для вирішення цієї проблеми запропоновано використовувати алгоритми на основі багатовимірних статистик. Дані алгоритми поєднують усі переваги статистичних методів та є єдиною альтернативою для аналізу послідовностей короткої та середньої довжини.

У статті наведено сумісні розподіли числа 2-ланцюжків і числа 3-ланцюжків фіксованого виду випадкової бітової послідовності. Можливим застосуванням отриманих формул може бути перевірка гіпотези випадковості розташування нулів і одиниць в (0, 1)-послідовний скінченної довжини.

Дослідження показало, що навіть при обмежених ресурсах та обмеженому середовищем, пов'язаному з ентропією, як смарт-карта, можна створити псевдовипадкові послідовності хорошої якості, які можуть задовольнити всі вимоги до ГПВЧ, навіть ті, які використовуються для комп'ютерів загального призначення. Метою роботи було розширити набір статистичних тестів, щоб включити інші тести, які не включені до статистичного набору NIST, і проаналізувати, чи реалізовані алгоритми їх задовольняють чи ні. В роботі наведено алгоритми для тестування ПВП з використанням багатовимірних статистик, щоб проілюструвати їх можливе застосування у середовищі смарт-карт.

ЛІТЕРАТУРА

1. **Овчинников А. И.** Тестирование датчиков случайных чисел, встроенных в смарт-карты. *Наука, техника и образование.* 2014. № 2.

2. **Rankl W.**, Effing W. Smart Card Handbook. New York, NY, USA: John Wiley & Sons, Inc., 2003.

3. **Akram R. N**, Markantonakis K., Mayes K. Pseudorandom Number Generation in Smart Cards: An Implementation, Performance and Randomness Analysis. *2012 5th International Conference on New Technologies, Mobility and Security (NTMS).* 2012. 10.1109/NTMS.2012.6208760.

4. **Koning Gans G.,** Hoepman J.-H., Garcia F. D. A Practical Attack on the MIFARE Classic. *CARDIS '08: Proceedings of the 8th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications*. Springer. 2008. Pp. 267–282.

5. **Garcia D.,** Koning Gans G., Muijrers R., Rossum P., Verdult R., Schreur R. W., Jacobs B. Dismantling MIFARE Classic," in ESORICS, 2008. Pp. 97–114.

6. **Nohl K**., Evans D., Starbug S., Plotz H. "Reverse-Engineering a ¨ Cryptographic RFID Tag," in SS'08: Proceedings of the 17th conference on Security symposium. Berkeley, CA, USA: USENIX Association, 2008. Pp. 185–193.

7. **Garcia D.,** Rossum P., Verdult R., Schreur R. W. "Wirelessly Pickpocketing a Mifare Classic Card," in SP '09: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE Computer Society, 2009. Pp. 3–15.

8. **Trichina E.,** Bucci M., Seta D., Luzzi R. "Supplemental Cryptographic Hardware for Smart Cards," IEEE Micro, vol. 21, no. 6, 2001. Pp. 26–35.

9. **Bucci M., Germani L., Luzzi R., Trifiletti A., Varanonuovo M**. "A High-Speed Oscillator-Based Truly Random Number Source for Cryptographic Applications on a Smart Card IC," IEEE Trans. Comput., vol. 52, no. 4, 2003. Pp. 403–409.

10. **Hambardzumyan E.**, Kim Y.-S., Karpins kyy B., "Fast Digital TRNG Based on Metastable Ring Oscillator," in Cryptographic Hardware and Embedded Systems – CHES 2008, ser. LNCS, E. Oswald and P. Rohatgi, Eds. Springer, August 2008, vol. 5154. Pp. 164–180.

11. **Biham E**., Shamir A. "Differential Fault Analysis of Secret Key Cryptosystems," in CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology. London, UK: Springer, 1997. Pp. 513–525.

12. **Boneh D**., DeMillo R. A., Lipton R. J. "On the Importance of Checking Cryptographic Protocols for Faults," in EUROCRYPT'97: Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques. Springer, 1997. Pp. 37–51.

13. **Common** "Criteria for Information Technology Security Evaluation", Part 1: Introduction and general model, Part 2: Security functional requirements, Part 3: Security assurance requirements, Common Criteria Std. Version 3.1, August 2006. [Online]. Available: http: //www.common criteriaportal.org /thecc.html.

14. "**BSI** AIS 31: Functionality classes and evaluation methodology for deterministic random number generators," Certification body of the BSI as part of the certification scheme version 2, 2001. [Online]. Available: https://www.bsi.bund.de/cae/ servlet/contentblob/ 478130/ publicationFile/30547/ais31epdf.

15. **FIPS** 140-2: Security Requirements for Cryptographic Modules. National Institute of Standards and Technology. Washington, DC. 2001.

16. **Kocher P.**, Jaffe J., Jun B., "Differential Power Analysis," Lecture Notes in Computer Science, vol. 1666, 1999. Pp. 388–397.

17. **Chari S.,** Rao J. R., Rohatgi P. "Template Attacks," in CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems. London, UK: Springer-Verlag, 2003. Pp. 13–28.

18. **Chari S.,** Diluoffo V. V., Karger P. A., Palmer E. R., Rabin T., Rao J. R., Rohatgi P., Scherzer H., Steiner M., Toll D. C. "Designing a Side Channel Resistant Random Number Generator," in Smart Card Research and Advanced Application, 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, D. Gollmann, J.-L. Lanet, and J. IguchiCartigny, Eds. Springer, April 2010. Pp. 49–64.

19. **BSI** AIS 20: Functionality classes and evaluation methodology for deterministic random number generators," Tech. Rep. version 2, December, 1999. [Online]. Available: https://www.bsi.bund.de/cae/ servlet/contentblob/478152/publicationFile/30552/ais20e pdf.pdf.

20. **NIST** Special Publication 800-57, Elaine Barker, William Barker, William Burr, William Polk, and Miles Smid «Recommendation for Key Management – Part 1: General (Revision 3)», July 2012.

21. **NIST** Special Publication 800-90A, Elaine Barker, John Kelsey, «Recommendation for Random Number Generation Using Deterministic Random Bit Generators», January 2012.

22. "**ISO**/IEC 18031: Information Technology-Security Techniques-Random bit generation," International Organization for Standardization and International Electrotechnical Commission, vol. iso 18031, 2005.

23. **Mayes K.,** Markantonakis K. "Smart Cards" Tokens, Security and Applications. Springer, 2008.

24. **Akram R. N.,** Markantonakis K., Mayes K. "A Paradigm Shift in Smart Card Ownership Model," in Proceedings of the 2010 International Conference on Computational Science and Its Applications (ICCSA 2010), B. O. Apduhan, O. Gervasi, A. Iglesias, D. Taniar, and M. Gavrilova, Eds. Fukuoka, Japan: IEEE Computer Society, 2010. pp. 191–200(eng)

25. **Кнут Д.** Искусство программирования, том 2. Получисленные методы / Д. Кнут. М.: Изд. дом «Вильяме», 2007.

26. **Brown R.** Dieharder: A Random Number Test Suite. [Online]. Available: http://www.phy. duke.edu/~rgb/ General/dieharder.php

27. **Security** Requirements For Cryptographic Modules. [Online]. Available: http://csrc. nist.gov/ publications/fips /fips140-2/fips1402.pdf.

28. **Popereshnyak S.,** Dimitrov G. "The Testing of Pseudorandom Sequences using Multidimensional Statistics" *Proceedings of the 1st International Workshop on Digital Content & Smart Multimedia* (DCSMart 2019) (Lviv, Ukraine, December 23–-25), 2019. P. 151–161.

29. **Masol V**., Popereshnyak S. Statistical analysis of local sections of bits sequence"s. *Journal of Automation and Information Sciences*. 2019. Vol. 51. P. 31-45.
DOI: 10.1615/JAutomatInf Scien.v51.i10.30

30. **Masol V**., Popereshnyak S. Checking the Randomness of Bits Disposition in Local Segments of the (0, 1)-Sequence. *Cybernetics and Systems Analysis.* 2020. Vol. 56(3). P. 1–8.
DOI: 10.1007/s10559-020-00267-0 (eng)

31. **Popereshnyak S.** The technique for testing short sequences as a component of cryptography on the Internet of Things. CEUR-WS.org/vol/ 2516/paper 11.

Поперешняк С. В.

ТЕСТУВАННЯ ДАТЧИКІВ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ, ВБУДОВАНИХ В СМАРТ-КАРТИ

У статті досліджено проблему випадковості та генерування її комп'ютерами загального призначення. Розглянуто вимоги безпеки, яким повинен задовольняти генератор псевдовипадкових чисел для використання в криптографічних програмах. Особлива увага приділяється відомим та новим методам тестування випадкових бітових послідовностей. Аналіз ефективності генераторів псевдовипадкових послідовностей є нагальною проблемою смарт-карт за умов використання більш досконалих методів шифрування та захисту інформації. Наявні способи показують низьку гнучкість та універсальність у засобах знаходження прихованих шаблонів у даних. Для вирішення цієї проблеми запропоновано використовувати алгоритми на основі багатовимірних статистик. Дані алгоритми поєднують усі переваги статистичних методів та є єдиною альтернативою для аналізу послідовностей короткої та середньої довжини. В статті розглянуто схему функціонування генераторів псевдовипадкових чисел в обмежених пристроях. Виділено основні вимоги до сучасних смарт-карт. Запропоновано критерій для перевірки на випадковість бітових послідовностей невеликої довжини (до 100 біт). Даний підхід доцільно використовувати для тестування полегшеного генератора псевдовипадкових чисел в пристроях з певними обмеженнями на ресурси. В роботі наведено сумісні розподіли числа 2-ланцюжків і числа 3-ланцюжків фіксованого виду випадкової бітової послідовності які дозволяють проводити статистичний аналіз локальних ділянок цієї послідовності. Можливим застосуванням отриманих формул може бути перевірка гіпотези випадковості розташування нулів і одиниць в (0, 1)-послідовний скінченної довжини. Дослідження показало, що навіть при обмежених ресурсах та обмеженому середовищем, пов'язаному з ентропією, як смарт-карта, можна створити псевдовипадкові послідовності хорошої якості, які можуть задовольнити всі вимоги до генераторів псевдовипадкових чисел, навіть ті, які використовуються для комп'ютерів загального призначення. В роботі було розширено набір статистичних тестів, щоб включити інші тести, які не включені до статистичного набору NIST, і проаналізувати, чи реалізовані алгоритми їх задовольняють чи ні. В роботі наведено алгоритми для тестування псевдовипадкової послідовності з використанням багатовимірних статистик, щоб проілюструвати їх можливе застосування у середовищі смарт-карт.

Ключові слова: смарт-карти; алгоритми; багатовимірної статистики; випадкові послідовності; s-ланцюжки; криптографія; псевдовипадкова послідовність; статистичне тестування.

Popreshnyak S.

TESTING PSEUDORANDOM NUMBER SENSORS BUILT IN SMART CARDS

This article explores randomness and how general purpose computers generate it. It also discusses the security requirements that a pseudo-random number generator must satisfy for use in cryptographic applications. And also special attention is paid to the known and new methods of testing random bit sequences. Analyzing the effectiveness of pseudo-random sequence generators is a pressing problem for smart cards when using more advanced methods of encryption and information protection. The available methods show low flexibility and versatility in the means of finding hidden patterns in data. To solve this problem, it is proposed to use algorithms based on multivariate statistics. These algorithms combine all the advantages of statistical methods and are the only alternative for analyzing short and medium-length sequences. The paper considers the scheme of operation of pseudo-random number generators in limited devices. The main requirements for modern smart cards are highlighted. A criterion for checking the randomness of bit sequences of small length (up to 100 bits) is proposed. This approach is appropriate for testing a lightweight pseudo-random number generator in devices with certain resource constraints. The paper presents the compatible distributions of the number of 2-strings and the number of 3-strings of a fixed form of a random bit sequence, which make it possible to carry out a statistical analysis of local sections of this sequence. A possible application of the obtained formulas can be to test the hypothesis of the randomness of the arrangement of zeros and ones in a (0, 1) - sequence of finite length. Research has shown that even with limited resources and a limited entropy environment like a smart card, good quality pseudo-random sequences can be created that can satisfy all the requirements for pseudo-random number generators, even those used for general purpose computers. In the work, the set of statistical tests was expanded to include other tests that are not included in the statistical set of NIST tests, and to analyze the work of the proposed algorithms. The paper presents algorithms for testing a pseudo-random sequence using multivariate statistics to illustrate their possible application in a smart card environment.

Keywords: Smart cards; algorithms; multidimensional statistics; random sequences; s-chains; cryptography; pseudorandom sequence; statistical testing.

Поперешняк С. В.

ТЕСТИРОВАНИЕ ДАТЧИКОВ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ, ВСТРОЕННЫХ В СМАРТ-КАРТЫ

Данная статья изучает случайность и как компьютеры общего назначения генерируют ее. В ней также рассматриваются требования безопасности, которым должен удовлетворять генератор псевдослучайных чисел для использования в криптографических приложениях. А также особое внимание уделяется известным и новым методам тестирования случайных битовых последовательностей. Анализ эффективности генераторов псевдослучайных последовательностей является насущной проблемой смарт-карт при использовании более совершенных методов шифрования и защиты информации. Имеющиеся способы показывают низкую гибкость и универсальность в средствах нахождения скрытых шаблонов в данных. Для решения этой проблемы предложено использовать алгоритмы на основе многомерных статистик. Данные алгоритмы сочетают все преимущества статистических методов и является единственной альтернативой для анализа последовательностей короткой и средней длины. В работе рассмотрено схему функционирования генераторов псевдослучайных чисел в ограниченных устройствах. Выделены основные требования к современным смарт-картам. Предложен критерий для проверки на случайность битовых последовательностей небольшой длины (до 100 бит). Данный подход целесообразно использовать для тестирования облегченного генератора псевдослучайных чисел в устройствах с определенными ограничениями на ресурсы. В работе приведены совместимые распределения числа 2-цепочек и числа 3-цепочек фиксированного вида случайной битовой последовательности, которые позволяют проводить статистический анализ локальных участков этой последовательности. Возможным применением полученных формул может быть проверка гипотезы случайности расположения нулей и единиц в (0, 1 –последовательности конечной длины. Исследования показало, что даже при ограниченных ресурсах и ограниченном средой, связанной с энтропией, как смарт-карта, можно создать псевдослучайные последовательности хорошего качества, которые могут удовлетворить все требования к генераторам псевдослучайных чисел, даже те, которые используются для компьютеров общего назначения. В работе был расширен набор статистических тестов, чтобы включить другие тесты, которые не включены в статистический набор тестов NIST, и проанализировать работу предложенных алгоритмов. В работе приведены алгоритмы для тестирования псевдослучайной последовательности с использованием многомерных статистик, чтобы проиллюстрировать их возможное применение в среде смарт-карт.

Ключевые слова: Смарт-карты, Интернет Вещей; алгоритмы; многомерные статистики; случайные последовательности; s-цепочки; криптография; псевдослучайная последовательность; статистическое тестирование.

Стаття надійшла до редакції 14.08.2020 р.
Прийнято до друку 21.09.2020 р.