КСЗ "Гриф-2" позволяет создать безопасную технологическую среду для систем электронного документооборота, банковских и других систем, для которых ключевым требованием является соблюдение конфиденциальности обрабатываемой информации и технологии ее обработки.

УДК 691.321

ПІДХОДИ ДО ВИЗНАЧЕННЯ ВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Геннадій Гулак, Людмила Ковальчук
ДСТСЗІ

Анотація: **Приводяться та порівнюються різні означення випадкових та псевдовипадкових послідовностей. Розглядається питання про тестування послідовностей.**
Summary: **Different definitions of random and pseudorandom sequences are compared. Question about testing of sequences is considered. Random and pseudorandom sequences, testing.**
Ключові слова: **Випадкові та псевдовипадкові послідовності, тестування.**

I Вступ

На практиці часто виникають два тісно пов'язаних між собою питання. Перше: як за поліноміальний час отримати послідовність чисел, що має певний набір властивостей, достатній для того, щоб її можна було вважати *випадковою*, тобто отриманою як *реалізацію послідовності незалежних випадкових величин з рівномірним розподілом,* що приймають значення у певному алфавіті? І друге питання: як визначити, чи можна деяку фіксовану послідовність вважати випадковою? У роботі розглядається декілька варіантів відповідей на ці запитання.

II Основна частина

§1. Визначення випадкової послідовності

Надалі ми розглядатимемо лише випадкові величини з дискретним розподілом, тобто для всіх *t* випадкові величини x_t приймають значення на множині $D = \{0, 1, ..., N-1\}$, що називається алфавітом послідовності.

За найбільш поширеним визначенням дискретна випадкова послідовність (РРВП) $\{x_t\}_{t\geq 1}$ – це послідовність незалежних (у сукупності) випадкових величин з рівномірним розподілом, тобто: $P(x_t = i) = N^{-1}$, де N називається об'ємом алфавіту D.

Випадкова послідовність має наступні властивості [11]:

1. $M(x_t) = 0.5 \cdot (N-1); \quad D(x_t) = 12^{-1} \cdot (N^2 - 1);$
2. $\forall k, \ \forall t_1, ..., t_k : \ P\{x_{t_1}, ..., x_{t_k}\} = N^{-k}$, тобто будь-який *k*-вимірний вектор з компонентами x_t має рівномірний на D^k розподіл.
3. будь-яка підпослідовність послідовності $\{x_t\}_{t\geq 1}$ також є РРВП;
4. сума за модулем *N* РРВП та будь-якої іншої послідовності, що від неї не залежить, також є РРВП;
5. РРВП не передбачувана, тобто для будь-якого натурального *n* кількість інформації по Шеннону, що міститься в $X_n = (x_1, ..., x_n)$ про майбутній елемент x_{n+1} дорівнює нулю: $I(x_{n+1}, X_n) = 0$.

Пристрій, що реалізує РРВП, називається *генератором* РРВП. Саме реалізації РРВП є тими об'єктами, що цікавлять нас. Одна з задач криптографії, що часто виникає на практиці, полягає у тому, щоб за конкретною реалізацією визначити, чи буде вона реалізацією РРВП. Для цього, зокрема, використовуються перелічені вище властивості РРВП та деякі інші її властивості. Надалі, для стислості викладу, реалізацію РРВП ми будемо називати просто випадковою послідовністю.

Існує багато визначень випадкової послідовності, але за різними причинами в своїй більшості вони не можуть бути застосовані на практиці.

Перший підхід до визначення випадкової послідовності був запропонований Шенноном та базується на теорії інформації [1]. Послідовність називається *випадковою за Шенноном*, якщо вміст інформації у ній

максимальний. Це можна сформулювати так: в послідовності відсутня надмірність; ентропія послідовності максимальна.

Якщо прийняти це визначення, то всі псевдовипадкові послідовності (генеровані генератором псевдовипадкових чисел – ГПВЧ), що досить часто використовуються на практиці, виявляться "поганими". Дійсно, оскільки вони отримані з більш короткого рядка – початкового заповнення, або "зерна" – то вони обов'язково містять надлишок інформації, а ентропія всієї послідовності дорівнює ентропії "зерна". Крім того, це означення стосується не окремої послідовності, а генератора, тобто імовірнісного розподілу на його виході.

Другий підхід був запропонований Колмогоровим [2]. Він базується на алгоритмічній складності обчислень. Послідовність довжини n називається *випадковою за Колмогоровим*, якщо її колмогорівська складність дорівнює n, тобто її не можна отримати в результаті роботи ніякого поліноміального алгоритму, довжина входу якого менша за n. Наприклад, колмогорівська складність послідовності з n одиниць дорівнює $\ln(n) + O(l)$, тому що її можна задати алгоритмом "надрукувати n одиниць", довжина входу якого $\ln(n)$. Подібно до шеннонівського означення, випадкова за Колмогоровим послідовність не може бути згенерована з більш короткого "зерна" поліноміальним алгоритмом, але, на відміну від нього, це означення може бути застосоване до окремої послідовності.

Третій підхід до визначення випадкової послідовності запропонували Блюм, Голдвассер, Мікалі та Яо [3,4]. У цьому випадку послідовність вважається випадковою, якщо не існує поліноміального (імовірнісного) алгоритму, який може відрізнити цю послідовність від суто випадкової. Послідовність, що має таку властивість, називається *поліноміально нерозрізнювальною з випадковою*, або *псевдовипадковою*.

Цей підхід дозволяє використовувати детерміновані алгоритми, що реалізуються деякими скінченими автоматами, для формування псевдовипадкових послідовностей максимального періоду [5]. Хоча з математичної точки зору така послідовність не є випадковою, оскільки вона повністю визначається початковим заповненням, вона може бути використана на практиці завдяки "нерозрізнювальності" з випадковою. Основні результати у цьому напрямку були отримані для бінарних послідовностей, але їх можливо узагальнити на випадок інших алфавітів. Оскільки цей підхід за визначенням уявляється більш конструктивним, зупинимося на ньому детальніше.

§2. Випадковість у сенсі практичного застосування

Послідовності, що є випадковими у сенсі третього означення, також називають "випадковими для всіх практичних застосувань." Генератори, що видають такі послідовності, називають криптографічно сильними (cryptographically strong) або криптографічно надійними (cryptographically secure). Псевдовипадковість у даному сенсі є не тільки властивістю послідовності (або генератора), але й властивістю спостерігача, а саме його обчислювальних можливостей.

У роботі [6] було доведено два важливих результати, що стосуються псевдовипадкових послідовностей.

1. Послідовність є псевдовипадковою тоді й тільки тоді, коли вона *непередбачувана*, тобто *витримує тестування наступним бітом* [7]. Це означає наступне. Якщо навіть нам відома частина послідовності будь-якої довжини, то (за умови невідомого початкового заповнення генератора та параметрів алгоритму генерації) для отримання її наступного біту ми не можемо запропонувати алгоритму, суттєво кращого за відгадування (або за підкидання монети).
2. Криптографічно сильні генератори існують у тому й тільки у тому випадку, якщо існують однобічні функції (one-way functions). У цьому випадку кожному псевдовипадковому генератору можна поставити у взаємно-однозначну відповідність деяку однобічну функцію, що залежить від певних параметрів.

Визначення випадкової послідовності за Блюмом-Голдвассером-Мікалі-Яо також не можна вважати практично бездоганним.

Насамперед, слід зауважити, що на даний час факт існування однобічних функцій не доведено, хоча на практиці широко використовуються так звані "кандидати в однобічні функції" – RSA-функція, функція дискретного логарифмування та функція Рабина. Інтуїтивно ясно, що до однобічних функцій слід віднести криптографічні алгоритми ГОСТ 28147-89, DES та відповідні хеш-функції.

По-друге, у зв'язку з швидким розвитком обчислювальної техніки, для збереження необхідних криптографічних якостей, а фактично – "однобічності" функцій, доводиться постійно збільшувати значення їх параметрів, що призводить до підвищення складності та тривалості не тільки зворотних, а й прямих обчислень.

І, по-третє, якщо нам треба зробити висновок про випадковість деякої послідовності, алгоритм генерації якої невідомий, як практично можемо визначити, чи існує поліноміальний алгоритм, що відрізняє її від випадкової?

Звернемо увагу на те, що значна кількість атак на криптографічні системи застосовує конкретні властивості ключової послідовності, зокрема, нерівномірний розподіл елементів ключів (інших критичних параметрів) або наявність статистичних залежностей між ними. Зокрема, в стандартах цифрового підпису DSS (США) та ГОСТ 34.310 (Росія, Україна) застосовується схема Ель-Гамаля, у якої цифровий підпис *(R, S)* розраховується за формулами:

$$R = \alpha^k \,(\mathrm{mod}\, p), S = (M \cdot k + x \cdot R) \,\mathrm{mod}(p-1)$$

де величини *R, S, M* вважаються відомими зловмиснику; невідомі *x* та *k,* відповідно, особистий ключ та деяка випадкова величина.

Якщо генератор випадкових чисел, що застосовується для формування величини *k*, не забезпечує рівномірний розподіл ймовірностей різних значень, то може стати реальною атака з частковим перебором найбільш ймовірних значень *k* у комбінації з наступним розв'язанням наведених рівнянь відносно особистого ключа.

У зв'язку з викладеним виникає необхідність, базуючись на підході Блюма-Голдвассера-Мікалі-Яо уточнити визначення випадкової послідовності таким чином, щоб його можна було застосовувати в разі тестування як певного генератора, так і окремої послідовності з урахуванням можливих атак на криптосистему, що побудовані на імовірнісно-статистичних властивостях відповідних послідовностей.

З урахуванням викладеного будемо вважати *послідовність символів певного алфавіту псевдовипадковою у вузькому сенсі, якщо ніякий алгоритм з визначеного набору поліноміальних алгоритмів не може її відрізнити від випадкової.*

Алгоритми з такого набору будемо називати тестами на випадковість, а процес перевірки послідовності – тестуванням. Якщо тест не зміг відрізнити послідовність від випадкової, то кажуть, що послідовність проходить, або витримує, даний тест, а тест приймає дану послідовність як випадкову. Навпаки, будемо говорити, що тест відхиляє послідовність.

Отже, питання про перевірку випадковості послідовності зводиться до питання про побудову набору тестів. Для цього спочатку потрібно визначити принцип їх роботи, а саме: за яким правилом тест має приймати одні послідовності і відхиляти інші?

Одним із способів вирішення цього питання є перевірка послідовності на наявність деякої властивості, притаманної майже всім випадковим послідовностям.

Розглянемо такий приклад. Нехай елементи послідовності вибираються незалежно й рівноймовірно з алфавіту {0,1}. Послідовність довжини *n*, що складається тільки з одиниць, навряд чи може вважатися випадковою для практичних застосувань, хоча й може бути одержана з ймовірністю 2^{-n}, як і будь-яка інша реалізація послідовності такої ж довжини незалежних, рівномірно розподілених на {0,1} випадкових величин.

Однак наведена послідовність не буде *типовою* в тому розумінні, що ймовірність того, що у реалізації РРВП достатньо великої довжини одиниць буде набагато більше, ніж нулів, досить мала. У той час уявляється доцільним вважати типовою послідовність, у якої кількості одиниць і нулів у ній відрізнятимуться несуттєво.

Тести, що спираються на таку властивість, називаються частотними, тому що вони фактично перевіряють розподіл частот символів алфавіту та виявляють його відхилення від рівноймовірного розподілу. До них відносяться, наприклад, критерії χ^2, максимальної правдоподібності та максимальної частоти. Крім частотних, існують тести, що перевіряють кількість серій (тобто груп, що складаються з однакових сусідніх елементів), кількість повторень певних груп символів, відстань між однаковими групами символів і т. д.

Одержавши результати тестування деякої послідовності невідомої природи, ми робимо висновки про її походження з досить високою ймовірністю (яка залежить від параметрів тесту), але меншою за 1. Результатом роботи тесту є прийняття або відхилення гіпотези H_0, яка полягає в тому, що дана послідовність є послідовністю незалежних, рівномірно розподілених випадкових величин. Альтернативна гіпотеза позначається H_2. Вона не може бути строго сформульованою, оскільки є складною. Вона охоплює як усі можливі нерівномірні розподіли елементів послідовності, так і усі можливі типи залежностей між ними.

З кожним тестом пов'язані два параметри: α – *ймовірність помилки 1-го роду* – ймовірність відхилити випадкову послідовність, тобто прийняти гіпотезу H_1 за умови, що вірна гіпотеза H_0, та β – *ймовірність помилки 2-го роду* – ймовірність прийняти послідовність, що не є випадковою, тобто прийняти гіпотезу H_0 за умови, що вірна гіпотеза H_1.

Величина α ще називається рівнем значимості тесту. Він задається на початку роботи тесту і дорівнює проценту суто випадкових послідовностей, що не мають властивості, наявність якої перевіряє тест. Зазвичай α обирається з практичних міркувань у залежності від відношення до гіпотези. Якщо гіпотеза, що перевіряється, є досить вірогідною, то рівень значимості обирається маленьким (10^{-2}–10^{-3}).

Параметр β залежить від α та довжини послідовності n і при фіксованому α зменшується з ростом n. Оскільки гіпотеза H_1 є складною, то з множини тестів ми не можемо обрати єдиний тест, який відповідає *рівномірно найбільш потужному критерію*, тобто такий, який при заданому рівні значимості α мінімізує значення β.

В ході роботи тесту обчислюється деяка величина U, що залежить від послідовності – так звана *статистика*, і тест приймає послідовність, якщо $U \le U_\alpha$ (або якщо $|U| \le U_\alpha$ – "двосторонній критерій"), де U_α – деяке граничне значення статистики, що залежить від α. Слід зазначити, що узгодження експериментальних даних з гіпотезою про наявність у послідовності певної властивості не свідчить про неможливість узгодження цих даних з іншою гіпотезою. При застосуванні статистичних критеріїв на основі спостережень неможливе доведення тієї чи іншої гіпотези. Можна лише стверджувати, що результати спостережень не суперечать даній гіпотезі.

Розглянемо основні вимоги до набору тестів.

1. Вибір необхідних тестів. Зрозуміло, що набір має містити деяку мінімально необхідну множину тестів, яка визначається майбутнім застосуванням тестованої послідовності та може бути різною для кожного конкретного випадку. Зокрема, до мінімального набору завжди має входити який-небудь частотний тест та тест, що аналізує розподіл серій.

Наприклад, набір тестів, рекомендований NIST для тестування двійкових послідовностей [8], містить 16 тестів (у т. ч. тести: частотний монобітний; частотний блочний; серій; довгих серій одиниць; рангу випадкової {0,1}-матриці; дискретного перетворення Фур'є; відповідності аперіодичних шаблонів, що не перекриваються; відповідності періодичних шаблонів, що перекриваються; універсальний статистичний – Маурера; Зіва-Лемпеля; лінійної складності; послідовності; наближеної ентропії; накопичених сум; випадкових відхилень; вигляду випадкових відхилень). Стверджується, що ці тести незалежні.

Більшість тестів з набору NIST розраховані на тестування двійкових послідовностей довжиною від 10^3 до 10^6, але деякі (монобітний та блочний частотні тести, тести серій, серій максимальної довжини, частот m-грам з перекриттям, наближеної ентропії та накоплених сум) допускають тестування послідовностей довжиною від 10^2 елементів.

Теоретично можливо розробити аналоги тестів цього пакету для тестування послідовностей елементів з довільного алфавіту, але при цьому виникають труднощі наступного характеру:

- обчислення відповідних параметрів тесту;
- збільшення довжини послідовності, необхідної для прийнятної вірогідності результатів тестуваня;
- збільшення необхідного об'єму пам'яті та кількості обчислень в ході роботи тесту, що в деяких випадках може призвести до значних витрат часу та ресурсів, необхідних для його виконання.

При практичному застосуванні вказаного набору можуть виникнути певні проблеми. По-перше, більшість тестів набору ефективні лише за умов досить великої довжини послідовності, яка підлягає тестуванню ($\ge 10^6$символів). По-друге, застосування окремих тестів потребує значних обчислювальних ресурсів через вельми великий час їх виконання (тести Маурера, Зіва-Лемпеля, лінійної складності). І, по-третє, параметри деяких тестів обчислені лише емпірично з використанням BBS або датчиків, реалізованих на основі хеш-функції типу MD-5 (тести Маурера, Зіва-Лемпеля). У §3 будуть приведені інші рекомендації щодо набору тестів для послідовностей різних довжин.

Окремо слід внести зауваження стосовно тесту Маурера [9]. Цей тест можливо застосовувати лише до досить довгих послідовностей ($n>10^6$). Крім того, за граничний розподіл тест Маурера використовує нормальний розподіл, хоча статистика є сумою величин, що не є незалежними. У своїй статті автор тесту уникає доведення збіжності розподілу статистики до стандартного нормального. Він називає свій тест "універсальним", виходячи з того, що він "здатний виявляти будь-який дефект з дуже загального класу статистичних дефектів", у той час, як, наприклад, критерій χ^2 виявляє лише частотні відхилення. Але авторам статті вдалося побудувати двійкову послідовність, що проходить тест Маурера при α=0,01 та довжині блоку L=8, але не проходить тест критерію χ^2. Довжина цієї послідовності 11 602 048 біт; статистика для тесту Маурера U=7,1795 при порогових значеннях 7,1671 та 7,2002; статистика для тесту критерію χ^2 U=675 743 при пороговому значенні 6,6349; крім того, вона має період 37 120 біт.

2. Незалежність тестів у сукупності. Для перевірки незалежності використовується n "еталонних" послідовностей (наприклад, отриманих з використанням однобічних функцій або хеш-функцій), де n має бути не менше за довжину послідовностей, до яких можна застосувати критерій незалежності χ^2 ($n\ge$500, а краще n=1000). Пропонується наступна емпірична методика перевірки незалежності тестів.

Нехай набір складається з n тестів $T_1,...,T_n$. Позначимо $T_1,...,T_n$ – події, що полягають в тому, що випадкова послідовність проходить тести $T_1,...,T_n$ відповідно. Потрібно перевірити, чи виконується рівність

$$P(\prod_{i=1}^{n} T_i) = \prod_{i=1}^{n} P(T_i).$$

За формулою множення ймовірностей

$$P(\prod_{i=1}^{n} T_i) = P(T_1 / T_2 \prod \ldots \prod T_n) P(T_2 / T_3 \prod \ldots \prod T_n) \ldots P(T_{n-1} / T_n) P(T_n).$$

Отже, для перевірки незалежності тестів у сукупності досить перевірити *n–1* попарну незалежність подій:

$$T_1 \text{ та } T_2 \cap \ldots \cap T_n;\ T_2 \text{ та } T_3 \cap \ldots \cap T_n;\ T_{n-2} \text{ та } T_{n-1} \cap T_n;\ T_{n-1} \text{ та } T_n.$$

Для цього обираємо достатню кількість "еталонних" послідовностей $\Sigma_1, \ldots, \Sigma_m$ та будуємо нові послідовності $M^{(i)},\ i = \overline{1, n}$, наступним чином:

$$M^{(1)} = (T_1(\Sigma_1), \ldots, T_1(\Sigma_m)), M^{(2)} = (T_2(\Sigma_1), \ldots, T_2(\Sigma_m)), \ldots, M^{(n)} = (T_n(\Sigma_1), \ldots, T_n(\Sigma_m)),$$

$$\text{де } T_i(\Sigma_j) = \begin{cases} 1, & \textit{якщо послідовність } \Sigma_j \textit{ проходить тест } T_i; \\ 0, & \textit{інакше} \end{cases}$$

Після цього будуємо послідовності $\Pi^{(k)}, k = \overline{1, n-1}$:

$$\Pi^{(1)} = M^{(1)}; \Pi^{(2)} = M^{(1)} \cdot M^{(2)}; \Pi^{(3)} = M^{(1)} \cdot M^{(2)} \cdot M^{(3)}; \ldots \Pi^{(n-1)} = M^{(1)} \cdot \ldots \cdot M^{(n-1)},$$

де під множенням послідовностей розуміють поелементне множення.

Тепер залишилось перевірити (попарну) незалежність наступних послідовностей:

$$\Pi^{(1)} \textit{ та } M^{(2)};\ \Pi^{(2)} \textit{ та } M^{(3)};\ \Pi^{(3)} \textit{ та } M^{(4)}; \ldots \Pi^{(n-1)} \textit{ та } M^{(n)}.$$

Незалежність перевіряємо, застосовуючи *n*–1 разів критерій χ^2 незалежності до перелічених пар послідовностей.

3. Визначення кількості тестів. Кількість тестів, що будуть входити до набору, визначається так званою *загальною помилкою першого роду*, тобто відносною кількістю (випадкових) послідовностей, які не будуть проходити хоча б один з тестів. Загальна помилка першого роду знаходиться за умови незалежності тестів при заданих помилках першого роду для кожного з тестів. Якщо для всіх тестів задана однакова помилка α (яку можна тлумачити як ймовірність того, що суто випадкова послідовність не пройде тест), то ймовірність того, що суто випадкова послідовність не пройде набір тестів (тобто не пройде принаймні один тест з набору), дорівнює

$$A = A(n, \alpha) = 1 - (1 - \alpha)^n = 1 - (1 - \alpha)^{\frac{1}{\alpha} n \alpha}$$

і при малих α приблизно дорівнює $1 - e^{-n\alpha}$.

Наприклад, якщо в кожному з 16 тестів, що входять до набору, рекомендованому NIST, задати помилку першого роду 0,01, то приблизно 15% суто випадкових послідовностей не буде проходити цей набір тестів.

Зауважимо, що з ростом *n* величина *A* зростає: $\lim_{n \to \infty} A(n, \alpha) = 1$, отже, кількість тестів не повинна бути надмірною. Якщо величина *A* задана, то кількість тестів визначається з рівняння $n = \left[-\alpha^{-1} \cdot ln(1 - A) \right]$.

§3. Щодо методик тестування послідовностей та генераторів

В рамках методики тестування випадкових послідовностей можливо виділити 3 основних напрямки:

- перевірка криптографічних якостей однієї конкретної послідовності "великої" довжини;
- перевірка криптографічних якостей однієї конкретної послідовності "малої" довжини;
- перевірка якості генератора, що використовується для отримання послідовностей.

Зупинимось спочатку на п. п. 1 і 2. Існує достатньо причин, щоб розглядати ці пункти окремо. Справа в тому, що властивості випадкової послідовності стають тим більш виразними, чим більша її довжина.

Як правило, вважається, що мінімальна довжина послідовності, для тестування якої використовуються частотні тести, у п'ять-десять разів більша від об'єму алфавіту. При такий умові імовірність того, що кожен символ зустрінеться у послідовності принаймні 1 раз (за умови рівномірного розподілу символів

послідовності) близька до одиниці. Дійсно, імовірність того, що якийсь символ не зустрівся у послідовності жодного разу, дорівнює $\left(1-\frac{1}{N}\right)^{n}=\left(1-\frac{1}{N}\right)^{10N}<e^{-10}\approx 5\cdot 10^{-5}$.

Для тестування іншими тестами довжина послідовності повинна бути ще більшою. Тому набір тестів, які можна запропонувати для тестування послідовності, суттєво залежить від її довжини і для коротких послідовностей буде невеликим.

Тестування послідовностей "великої" довжини

Виходячи з відомих атак на криптографічні системи, уявляється доцільним застосування наступного набору критеріїв для тестування послідовностей символів з алфавіту об'єму N достатньо великої довжини n:

КД1. Критерій χ^2 – для перевірки гіпотези щодо рівноймовірного розподілу символів.

При цьому слід зауважити, що при $n>10N^{\nu}$ для деякого $2\le\nu\le16$, можливо краще виконувати тестування критерієм χ^2 для ν-грам для максимально допустимого ν. При виборі ν>16 виникають проблеми зі складністю та часом обчислень навіть для двійкового алфавіту.

КД2. Критерій серій максимальної довжини;

КД3. Критерій кількості серій.

Ці критерії виявляють міжсимвольні залежності протилежної природи. Перший з них виявляє таку залежність, коли ймовірність появи символу, що дорівнює попередньому, значно більша, ніж будь-якого іншого, а другий – коли така ймовірність менша. Разом з ними для більш детального аналізу розподілу серій можливо використовувати

КД4. Критерій кількості *S*-серій,

який аналізує розподіл серій всіх довжин.

КД5. Критерій інверсій, або перевищень,

виявляє монотонні залежності між символами послідовності, тобто такі залежності, коли ймовірність виникнення у послідовності відрізку великої довжини, близького до монотонного, більша, ніж довільного відрізку.

КД6. Критерій місць знаків

виявляє залежність символів послідовності від їх місць, наприклад, якщо ймовірність появи деяких символів у першій половині послідовності суттєво відрізняється від ймовірності їх появи у другій.

Перелічені вище критерії не виявляють досить широку множину послідовностей з коротким періодом, наприклад, послідовності типу 111222333...999000111222...000111... і т. д. Тому для подальшого аналізу послідовностей необхідно використати групу тестів, які виявляють існування підпослідовностей, розподіл елементів яких суттєво відрізняється від рівномірного:

КД7. Критерій наявності статистичного періоду;

КД8. Критерій наявності статистичного збою;

КД9. Критерій наявності статистичного зсуву.

Кожен з них виявляє відхилення розподілу у підпослідовності, що вибрана за певним правилом (своїм для кожного тесту). Наприклад, приведена нами послідовність буде відбракована критеріями наявності статперіоду та статзбою.

На наш погляд, тестування послідовності "в цілому" може бути недостатнім. Оскільки (§1, пп. 3), будь-яка підпослідовність РРВП – є також РРВП, наряду з тестуванням всієї реалізації повинні виконуватись тестування її різних підпослідовностей. Ці підпослідовності повинні обиратись за певними правилами, так, щоб символ, який обирається не залежав від свого значення. Зокрема, не можна обирати підпослідовність, що складається з усіх нульових елементів послідовності. Правила вибору послідовності можуть бути наступними:

- кожен k-ий елемент послідовності;
- кожен елемент, перед яким стоїть деяка фіксована комбінація символів;
- елементи, номери яких визначаються деяким пуасонівським законом;
- елементи, номери яких визначаються деякою іншою послідовністю, незалежною від даної.

Тестування послідовностей "малої" довжини

Для таких послідовностей здебільшого використовують невеликий набір тестів:

КК1. Критерій χ^2 або χ^2 для ν -грам, якщо довжина послідовності більша за $10N^{\nu}$;

КК2. Критерій серій максимальної довжини;

КК3. Критерій кількості серій.

Для двійкової послідовності критерій кількості серій співпадає з так званим критерієм знакозмін.

КК4. Критерій інверсій;

КК5. Критерій місць знаків.

Тестування генераторів псевдовипадкових чисел

Тестування генераторів складається з двох етапів:

- тестування достатньої кількості ($\approx 10^3$) послідовностей відібраними тестами;
- обробка результатів.

Як було зазначено раніше, при застосуванні тесту порівнюються значення U та U_α. Величина U_α називається пороговим значенням статистики і знаходиться зі співвідношення $\Phi(U_\alpha) = 1 - \alpha$, де Φ – функція розподілу ймовірностей, α – рівень значимості тесту. Оскільки функція Φ – зростаюча, замість порівняння U і U_α можна порівнювати $\Phi(U)$ та $\Phi(U_\alpha)$. Значення $1-\Phi(U)$ називається P-величиною і позначається P_v. Отже, послідовність визнається випадковою, якщо $P_v \geq \alpha$.

Величини α і P_v є ймовірностями того, що суто випадкова послідовність буде мати статистику, більшу за U_α та U відповідно. Отже, для генератора суто випадкових послідовностей має виконуватись наступне твердження: якщо ми тестуємо послідовності, які він видає, деяким тестом з заданим рівнем α, то з ймовірністю α послідовності не будуть проходити цей тест. Наприклад, при α=0.01 приблизно 1 з 100 послідовностей не буде проходити тест. Також треба відмітити той факт, що генератор суто випадкових послідовностей повинен генерувати послідовності так, щоб відповідні їм значення P_v були рівномірно розподілені.

NIST пропонує методику обробки результатів тестування, що складається з 2 пунктів:

1. Обчислення пропорції послідовностей, що пройшли тест:

Нехай було протестовано *m* послідовностей, і з них *k* пройшли тест. Тоді пропорція дорівнює *k/m.* Якщо рівень значимості тесту дорівнює *α,* має виконуватись нерівність:

$$1 - \alpha - 3\sqrt{\frac{\alpha(1-\alpha)}{m}} \leq \frac{k}{m} \leq 1 - \alpha + 3\sqrt{\frac{\alpha(1-\alpha)}{m}}. \qquad (1)$$

2. Перевірка рівномірності розподілу P-величин:

Обчислюється статистика

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - 0.1 \cdot m)^2}{0.1 \cdot m},$$

де F_i – кількість P-величин, що належать до інтервалу $(0.1 \cdot (i-1), 0.1 \cdot i)$, $i = \overline{1,10}$.

Обчислюється значення $P_T = igams(4.5, 0.5 \cdot \chi^2)$, де *igams* – неповна гама-функція; вона обчислюється за формулою:

$$\Theta(a, x) = \frac{1}{\Gamma(a)} \int_{[}^{\infty} t^{a-1} e^{-t} dt,$$

де

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt.$$

Якщо виконується нерівність

$$P_T \geq 0.0001, \qquad (2)$$

то P-величини вважаються рівномірно розподіленими.

Генератори, для яких виконуються (1), (2), за термінологією NIST [10] називаються *схваленими* (*approved*).

Виходячи з означення загальної помилки першого роду, пропонується доповнити методику ще одним пунктом.

3. Обчислення пропорції послідовностей, що пройшли всі тести:

Нехай було протестовано *m* послідовностей, і з них k пройшли всі тести. Тоді пропорція дорівнює *k*/*m*. Якщо загальна помилка першого роду дорівнює A, має виконуватись нерівність:

$$1 - A - 3\sqrt{\frac{A(1-A)}{m}} \le \frac{k}{m} \le 1 - A + 3\sqrt{\frac{A(1-A)}{m}}\ . \qquad (3)$$

Якщо виконуються нерівності (1), (2), (3), то генератор будемо вважати генератором суто випадкових послідовностей.

Якщо такий генератор використовується у криптографічних модулях, до яких висуваються вимоги 3 – 4 рівнів безпеки [10], то NIST рекомендує також проводити тестування генераторів при запуску (рівень 4) та при необхідності в будь-який інший момент (рівні 3, 4). Тестування полягає у застосуванні до перших 20 000 бітів згенерованої послідовності наступних тестів (при α=0,001):

- монобітний тест (тобто критерій χ^2);
- тест покеру (критерій χ^2 для 4-грам, що не перекриваються);
- тест серій (критерій кількості *S*-серій);
- тест серій максимальної довжини.

Крім того, протягом роботи генератора постійно має виконуватись наступний тест. Починаючи з 1-го біту, згенерована послідовність розбивається на *n*-грами ($n \ge 16$). Кожна наступна *n*-грама не повинна дорівнювати попередній.

Література: 1. T. M. Cover, G. A. Thomas, Elements of information theory, Wiley, New-York, 1991. 2. M. Li, P. Vitanyi, An introduction to Kolmogorov complexity and its applications, Springer-Verlag, New-York, 1993. 3. M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, SIAM J. Comput. 13(1984), p. 850–864. 4. A. C. Yao, Theory and applications of trap-door functions, 23rd IEEE Symposium on Foundations of Computer Science, 1982, pp. 80–91. 5. Л. В. Ковальчук, О периодах выходных последовательностей некоторых псевдослучайных генераторов, построенных на основе односторонних функций, III Международная научно-практическая конференция "Безопасность информации в информационно - телекоммуникационных системах", Киев, 2000. 6. O. Goldreich, Pseudorandomness, Notices of AMS, v. 46, № 10, 1999, p. 1209–1216. 7. О. Вербицький, Вступ до криптології, Видавництво науково-технічної літератури, Львів, 1998, 247 с. 8. NIST Special Publications 800–22, A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications, 2000. 9. U. M. Maurer, A Universal Statistical Test for Random Bit Generators, Journal of Cryptolojy, № 5, 1992, p. 89–105. 10. Federal Information Processing Standard Publications "Sequrity Requirements for Cryptographic Modules" 140–1, January 11, 1994, 50 p. 11. Ю. С. Харин, В. И. Берник, Г. В. Матвеев, Математические основы криптологии, Минск, БГУ, 1999, 319 с.

УДК 681.3.06

ПОСТРОЕНИЕ ЛИНЕЙНЫХ БАЗИСОВ ВЕКТОРНЫХ ПРОСТРАНСТВ НАД ПОЛЕМ GF(2)

Сергей Головашич

Харьковский национальный университет радиоэлектроники

Анотація: **Пропонуються два алгоритми побудови лінійних базисів над простором $GF(2)^n$. Перший алгоритм дозволяє здійснювати вибір з повної множини базисів над $GF(2)^n$, але є відносно складним. Другий є значно простішим, але здійснює вибір не з повної множини.**

Summary: **Two algorithms for building a linear basis over $GF(2)^n$ are proposed. The first one allows choosing a linear basis of the full space, but it is comparatively complicated. The second one is much easier, but it performs choosing from the truncated space.**

Ключевые слова: **Криптография, линейное отображение, базис линейного пространства, невырожденная (не сингулярная) матрица.**

Процедуру блочного симметричного шифрования (БСШ) можно рассматривать как композицию множества элементарных отображений всего блока данных либо его фрагментов. Эти элементарные отображения можно разделить на три группы:

- фиксированные – отображения, постоянные для всех блоков данных и ключей шифрования (например, структурные схемы SPN или SLTN-преобразований, нелинейные преобразования – S-блоки);