УДК 519.6

Ю.П. Буценко, Ю.Г. Савченко

АВТОМАТНІ МОДЕЛІ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ БІНАРНИХ ЧИСЕЛ

Розглянуто задачу формування послідовностей псевдовипадкових бінарних чисел за допомогою скінченних цифрових автоматів моделі Мілі. У порівнянні з традиційними реалізаціями генераторів на регістрах зсуву із зворотними зв'язками по модулю 2 використання такої універсальної моделі потенційно може покращити статистичні показники якості генерованих послідовностей за критерієм складності алгоритму генерації, запропонованим А.М.Колгоровим, та критеріями, що застосовуються при оцінках при використанні тестів NIST.

Ключові слова: псевдовипадкові числа, генератори послідовностей, тестування на випадковість.

Вступ

Використання послідовностей псевдовипадкових бінарних чисел (ПВБЧ) досить різноманітне та поширене. Мова йде передусім про криптографічний та стеганографічний захист інформації, сигнатурне діагностування цифрових пристроїв, керування несівними частотами в сучасних телекомунікаційних мережах тощо. У криптографії послідовності ПВБЧ – це ключі для шифрування (дешифрування), які необхідно змінювати досить часто (в деяких сучасних міжнародних стандартах навіть для кожного сеансу зв'язку), а при використанні потокових шифрів за схемою "скремблер-дескремблер" генератор ПВБЧ, по суті, стає центральною складовою процедури захисту. У багатьох випадках також застосовують так зване "забілювання" інформаційного потоку шляхом накладання на нього послідовності ПВБЧ з метою маскування статистичних особливостей потоку.

Іншою областю використання ПВБЧ є сигнатурні аналізатори, де основним компонентом є генератор ПВБЧ, що формує вхідні тестові сигнали на цифровий пристрій (об'єкт контролю) для визначення наявності несправності.

У всіх зазначених випадках генерована послідовність ПВБЧ повинна бути максимально наближеною (подібною) до "справжньої" випадкової, оскільки ступінь подібності визначає фактично рівень захищеності інформаційного обміну або достовірність отриманого результату діагностування.

Вимоги до довжини послідовності ПВБЧ в залежності від конкретного застосування можуть бути різними. Так, в системах скремблер-дескремблер бажано отримати послідовність максимально можливої довжини (принаймні, того ж порядку, що довжина відкритого тексту), а для ключів в стандартних шифрах (DES,FES) це 64…512 біт.

Основна частина

Історично першими були реалізації генераторів псевдовипадкових бінарних чисел (ГПВБЧ) на основі регістрів зсуву із зворотними зв'язками по модулю 2 або LFSR (Linear feedback shift register). Це пов'язано із обмеженнями функціональних можливостей елементної бази того часу. Зокрема, важливим фактором були (і залишаються) масо-габаритні обмеження при апаратній реалізації та необхідність вбудовування генераторів у портативну апаратуру зв'язку та пристрої спеціального призначення. На сьогодні ситуація докорінно змінилася насамперед з точки зору можливостей сучасної мікроелектроніки. Очевидно, що, застосовуючи для побудови ГПВБЧ компоненти із практично необмеженими функціональними можливостями (в межах детермінованих перетворень), можна сподіватися на створення більш досконалих ГПВБЧ. Зазначимо зразу ж, що поняття «досконалість ГПВБЧ» або, простіше, «якість» досить розпливчате та визначається якістю *послідовності генерованих чисел.* На змістовному рівні якість можна оцінювати мірою наближеності

заданої послідовності до ідеальної, тобто такої, в якій її фрагменти довільної довжини з'являються в послідовності з однаковою частотою (мають рівномірний розподіл).

Один із відомих тестів на випадковість є «тест на наступний біт» – процедура для перевірки псевдовипадкових послідовностей на криптостійкість. Ідея полягає в тому, що не повинно існувати поліноміального алгоритму, який би міг на основі перших k біт послідовності спрогнозувати $k+1$ біт з ймовірністю, більшою $1/2$. Ендрю Яо у 1982 році довів, що генератор, який пройшов тест на наступний біт, пройде також будь-які інші статистичні тести на випадковість, що можуть бути виконані за поліноміальний час.

Ще один із тестів [1, 2, 3] для оцінки якості використовує критерій *складності алгоритму генерації*. На думку авторів, складність будь-якого алгоритму чисельно (об'єктивно) оцінити проблематично, навіть, якщо обмежитись апаратною його реалізацією на регістрах зсуву. Зупинимось на цьому питанні детальніше.

Можна висловити таке (можливо дискусійне) міркування. Відповідно до принципу Кірхгофса, умовному криптоаналітику (противника), відомо, будемо вважати, все, окрім ключа. А сама бінарна псевдовипадкова послідовність є об'єктом його аналізу – це ключ для розшифрування. У нашому випадку криптоаналітик «знає» клас (тип) апаратури, що застосовується для генерації. У традиційному випадку – це LFSR. Повний опис такого генератора вичерпується описом конкретного виду зворотних зв'язків в регістрі та n–розрядним стартовим словом, з якого починається генерація. Загалом, цей опис має об'єм $2n$ біт, де n довжина регістра. Зважаючи, що реальні значення n лежать в межах 32…64 біт, опис конкретного генератора є доволі коротким. Формально це може свідчити про невисоку умовну «якість» відповідного генератора (якщо вважати, що якість еквівалентна складності алгоритму генерації та його опису).

Аналогічний критерій був запропонований А.М.Колгоморовим[4,5], відповідно до якого якість генерованої послідовності визначається, суттєво спрощуючи питання, *довжиною опису* алгоритму (процедури) генерації. Такий підхід значною мірою є гіпотетичним, оскільки існують приклади алгоритмів, коли при короткому описі генерується послідовність відносно великої довжини із прийнятними статистичними характеристиками.

Необхідно також окремо зазначити, що обов'язково повинна бути можливість повторити генерацію *точно тієї ж самої* конкретної послідовності ПВБЧ скільки завгодно раз. Така вимога визначена, власне, основним призначенням генераторів. При захисті інформаційного обміну сформована послідовність – це ключ шифрування, а при дешифруванні отримавувачу повідомлення необхідно в більшості випадків використати точно такий же ключ, що й при шифруванні. При сигнатурному діагностуванні принципово важливо формувати вихідні еталонні сигнатури та проводити діагностування з використанням однакових тестових послідовностей. Тому використання генераторів *дійсно* випадкових чисел (наприклад, таких, що базуються на квантуванні шумових сигналів) є принципово неприйнятним.

Перейдемо безпосередньо до розгляду моделей формування послідовностей ПВБЧ та спробуємо кількісно оцінити складність алгоритму генерації у випадку його апаратної реалізації. Тоді як узагальнену модель ГПВБЧ можна застосувати модель скінченних цифрових автоматів (рис. 1).

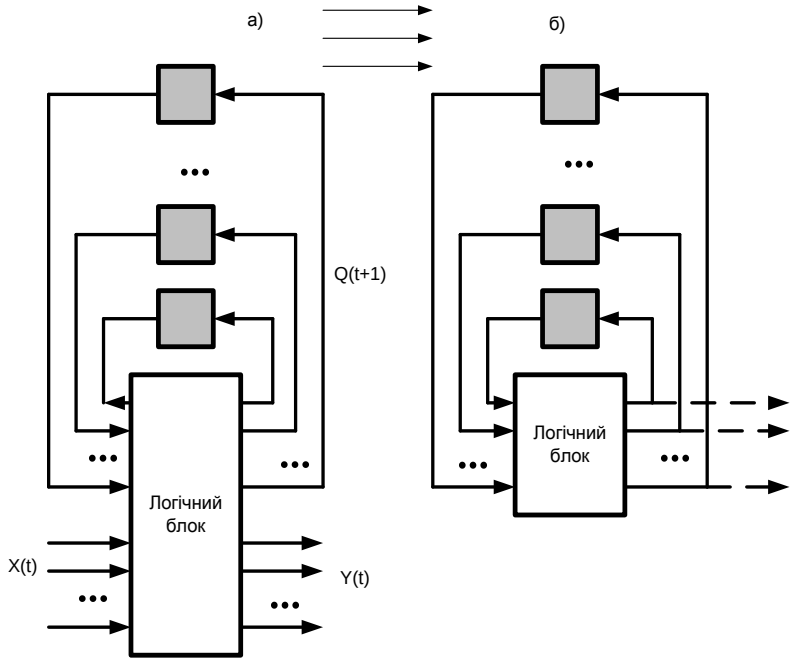

Рис. 1. Узагальнена схема генератора наоснові автоматних моделей (а – модель Мілі; б – модель Мура)

Однак, якщо вважати генератор автоматом «без входів», то можна використати й простішу модель, а саме модель автомата Мура (рис. б), де квадратні елементи − це однобітові елементи пам'яті або затримки на один період тактових імпульсів, а $X(t), Y(t), Q(t)$ − двійкові вектори довжиною l, m, n відповідно. У цьому випадку виходами генератора стають фактично сигнали, які відображають внутрішній стан автомата. Тоді

$$q_i(t+1) = \varphi_i[q_1(t), q_2(t), ...q_m(t)],\ i = 1, 2, ...m,$$

де $q_i(t)$ та $q_i(t+1)$ − значення поточного та наступного стану i-го елемента пам'яті генератора, а φ_i − відповідна булева функція переходу.

Іншими словами, кожен наступний стан автомата *однозначно* залежить лише від попереднього стану, а вся послідовність генерованих чисел однозначно визначається графом переходів автомата. У випадках, коли бажано отримати послідовність максимальної довжини $L_{\max}$, автомат повинен пройти через усі свої стани. Очевидно, що при відсутності вхідних впливів граф переходів буде 1-зв'язаним і не мати циклів, а $L_{\max} = n2^n$ біт. Це випливає безпосередньо із однозначності переходу з поточного стану в наступний. Тобто у графі переходів кількість дуг, що виходять з кожної вершини, дорівнює в точності 1. Інших варіантів не існує. У випадку, коли деякі вершини є недоступними, довжина генерованої послідовності $L < L_{\max}$. Очевидно, для утворення інших траєкторій необхідно створити умови для зовнішнього керування, тобто на генератор подавати ще й вхідні сигнали (впливи) та перейти до загальної моделі Мілі. У цьому випадку для визначення поведінки автомата необхідно додатково задати ще й його функції виходів

$$y_j(t) = f_j[x_1(t), x_2(t), ...x_l(t); q_1(t), q_2(t), ...q_m(t)],\ j = 1, 2, ...l,$$

де f_j − вихідні булеві функції автомата (генератора).

В залежності від різноманіття таких впливів можна створити більшу чи меншу кількість траєкторій зміни внутрішніх станів генератора. Очевидно, максимальна кількість цих траєкторій визначається кількістю входів автомата l і дорівнює 2^l. Якщо $l = 1$, то генератор може видати дві послідовності чисел (по закінченні однієї згенерувати другу). А це вже прямий шлях до суттєвого збільшення довжини генерованої послідовності.

До речі, саме цим шляхом можна вдосконалити генератори, побудовані на базі LFSR. Відомо [6], що для отримання послідовності максимальної довжини в таких генераторах зворотні зв'язки необхідно вибирати відповідно до степеней примітивних поліномів, які є дільниками полінома виду $X^{2^n} \oplus 1$, де n – довжина регістра. Для кожного значення n існує деяка сукупність таких неприводимих поліномів, Так, наприклад, для $n = 10$ це вже десятки поліномів, за допомогою яких можна згенерувати послідовності максимальної довжини. Тобто, заклавши в генератор можливість програмної зміни зворотних зв'язків у регістрі, можна у десятки разів подовшити генеровану послідовність[7].

Але головне: у цьому випадку зникає однозначність переходу від одного стану автомата до іншого, що збільшує шанси успішно пройти тестування деякими статистичними тестами NIST, зокрема такі тести єнайрозповсюдженішими програмами тестування послідовностей. Зокрема, у відповідності до [3] вимоги до послідовності можуть бути підсумовані у чотирьох пунктах:частотна стійкість, хаотичність, типовість та непередбачуваність. Щодо першої з цих вимог, то так звана стохастичність за А.М.Колгоморовим (удосконалення стохастичності за Черчем) означає розгляд лише такого класу S послідовностей, для яких всі підпослідовності мають властивість статистичної стійкості, тобто відносна частота появи в послідовності 0 або 1

$$\nu(i) \to 1/2 \text{ при } i \to \infty,$$

де i – кількість фрагментів послідовності, для яких ця ймовірність вимірюється.

Використання моделі функціонування генератора як автомата Мілі дозволяє забезпечити проходження тестів I…VIII, X, XI із сукупності тестів NIST.

Щодо хаотичності (за А.М.Колгоморовим характеризується складністю опису алгоритму генерації), то це питання розглянуто нижче і для пропонованої моделі в значній мірі залежить від можливостей мінімізації вибраних булевих функцій виходів і переходів. Але оцінити чисельно ці можливості у випадку довільних булевих функцій проблематично, тому для хоча б орієнтовної оцінки хаотичності використовують тестування на рівень можливого стиснення послідовності при використанні стандартних алгоритмів компресії (тести IX, XIV сукупності тестів NIST).

Типовість послідовності (тести XII, XIII), мабуть, є найскладніших критерієм для чисельної оцінки. Можна лише стверджувати, що для успішного проходження тестів на типовість послідовність повинна бути досить довгою та такою, що не утворена простою перестановкою відносно невеликої кількості коротких фрагментів.

Зазначимо нарешті, що непередбачуваність послідовності, тобто неможливість передбачити наступний біт на основі всіх попередніх значень з ймовірністю, більшою за$1/2$ вже згадувалась раніше. Важливо пам'ятати, що непередбачувана послідовність є стохастичною за А.М.Колмогоровим, але цей клас є істотно ширший за клас непередбачуваних. Питання ж про співвідношення між цими класами сформульоване в термінах теорії ігор, залишається наразі однією з найактуальніших її проблем.

Але повернемось до загального випадку, а саме використання моделі Мілі (рис. 1,а). Наступним кроком в пошуку нових варіантів побудови ГПВБЧ може бути перехід до довільних логічних функцій переходів. Зазначимо, що в класичній схемі при використанні LFSR це можуть бути лише лінійні логічні функції, а функції переходів мають зазделегідь фіксований вигляд

$$q_1(t+1) = \oplus \sum_{\lambda \in M} q_\lambda(t),$$
$$q_2(t+1) = q_1(t),$$
$$q_3(t+1) = q_2(t),$$
$$.....$$
$$q_n(t+1) = q_{n-1}(t),$$

де $\oplus \sum_{\lambda \in M} q_\lambda(t)$ – сума по модулю 2 бітів у відповідності до вибраного поліному.

Більш детальний та глибокий аналіз показує, що при використанні інших (нелінійних) функцій неможливо отримати послідовність максимальної довжини. Але, з іншого боку, можна отримати широкі можливості варіацій статистичних характеристик.

Якщо скористатися всіма можливостями узазальненої моделі Мілі, виникає питання: які вхідні сигнали подавати на генератор? Де їх взяти та хто їх має створювати? Тобто модель необхідно доповнити деяким блоком керування, який буде виконувати ці функції, формуючи відповідні вхідні сигнали

$$X(t) = [x_1(t), x_2(t), ..., x_l(t)].$$

Для спрощення припустимо, що вхідні сигнали – це числа (наприклад, від двійкового лічильника), які задають порядок в часі тієї чи іншої програми генерації послідовності. Тобто введення вхідних керуючих впливів дозволяє суттєво збільшити довжину генерованої послідовності. Але це, мабуть, не головне. Найбільш важливим, на наш погляд, є комбінаторне збільшення різноманіття можливих послідовностей за рахунок зміни функцій виходів і переходів автомата.

Дійсно, якщо розглядати проблему якості генератора та генерованих за його допомогою послідовностей не суто формально, а на змістовному рівні, то проблема еквівалентна відповіді на досить просте питання: легко чи важко криптоаналітику знайти алгоритм, за яким утворена послідовність? При використанні пропонованої моделі генератора криптоаналітику потрібно буде зробити вибір із $N = 2^{2^{n+l}}$ варіантів лише для функцій виходів. Нескладний розрахунок показує, що вже для невеликих значень n та l перебір варіантів практично не може бути реалізований за часовими витратами навіть за допомогою суперкомп'ютера. Так, наприклад, при $n + l = 16$ кількість варіантів стає більшою за 10^{3000}. Якщо навіть відкинути ті логічні функції, які за визначенням є непридатними (константи 0 та 1, вироджені функції тощо), то все одно перебір залишається таким, який не можна реалізуватипрактично. Тобто є підстави вважати, що шанси знайти закономірності, за якими побудована генерована послідовність, практично нульові.

Можна розмірковувати й трохи по-іншому. Спираючись на, без сумніву, фундаментальний постулат А.М. Колмогорова про еквівалентність складності псевдовипадкової послідовності складності опису алгоритму (процедури) її формування, можна припустити, в свою чергу, що складності опису алгоритму відповідає довжина його опису, наприклад, у бінарних символах. (Звичайно, такий підхід є спрощеним, але дозволяє отримати хоча б орієнтовні кількісні оцінки). Виходячи з такої точки зору, спробуємо оцінити довжину опису алгоритму створення *конкретної* псевдовипадкової послідовності.

По-перше, слід задати n булевих функцій переходів та m функцій виходів. Тобто загалом опис генератора повинен містити $(m+n)$ описів конкретних булевих функцій від $(n+l)$ двійкових змінних, а це їх таблиці істинності. Кожна така таблиця може бути описана множиною наборів, на яких вона дорівнює 1 (або еквівалентного запису "по нулям"), а довжина опису для однієї функції складатиме максимум 2^{n+l-1} бітів. Для всіх функцій генератора це буде

$$(m+n)2^{m+n-1} \text{ біт.}$$

До цієї величини необхідно додати величину вектора ініціалізації, тобто початковий внутрішній стан автомату та стартове слово, з якого починається генерація. Це ще $(m+l)$ біт.

Таким чином, повний опис алгоритму генерації конкретної послідовності при використанні універсальної автоматної моделі складає

$$M = m + l + 2^{n+l-1}(m+n) \text{ біт.}$$

При апаратній реалізації генератора та типових значеннях $m = n = l \approx 12...16$ обсяг опису складатиме $10^{10}...10^{12}$ біт у порівнянні із реалізацією на LFSR, де він складає 24…32 біти.

Висновки

На завершення слід зазначити, що використання обсягу опису алгоритму генерації послідовності ПВБЧ для визначення якості послідовності залишається дискусійним і на практиці не конкурує із використанням тестів NIST. В той же час, виходячи з принципу Кірхофса, згідно з яким криптоаналітику відома загальна схема алгоритму генерації, вже на інтуїтивному рівні очевидно, що трудомісткість криптоаналізу при застосуванні LFSR та, у другому варіанті при гіпотетичному використанні скінченого цифрового автомата моделі Мілі розрізняються на порядки. Так у першому випадку можуть бути застосовані досить ефективні алгебраїчні атаки, наприклад [6], а у другому − скоріш за все, лише перебір логічних функцій та їх комбінацій. Звичайно, множини функцій виходів і переходів, які доцільно використовувати в конкретних реалізаціях генераторів, потребують додаткового моделювання та перевірки на придатність, але кількість можливих варіантів навіть після такої «фільтрації» залишається песимістичною з точки зору часових витрат.

Література

1. Потий А., Орлова С. Статистическое тестирование генераторов случайных и псевдослучайных чисел с использованием набора статистических тестов NISTSTS. / А. Потий, С. Орлова. – Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні. – 2001, вип. 2. – С. 206–214.

2. Andrew Rukhin, NIST Statistical Test Suite.csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html

3. Statistical Testing of Random Number Generators / Proceedings of the 22-nd National Information Systems Security Conference, – 10/99.

4. Колмогоров А.Н. Три подхода к определению понятия «количество информации» / А.Н.Колмогоров // Проблемы передачи информации, 1964 ,Т.1(1). – С. 3–11.

5. Вьюгин В.В. Колгоморовская сложность и алгоритмическая случайность / В.В. Вьюгин. – М.: МФТИ, 2012. –131 с.

6. Берлекэмп Э. Алгебраическая теория кодирования / Э. Берлекэмп. – М.: Мир, 1971. – 477 с.

7. Малогулко Р.В, Вдосконалення генераторів ПВП та їх застосування в системах скремблер-дескремблер телекомунікаційних пристроїв // Р.В.Малогулко, Ю.Г.Савченко // Наукові записки УНДІЗ, 2008, випуск № 6(8), С.43–49.

8. Пометун С.О. Алгебраїчні атаки на потокові шифратори як узагальнення кореляційних атак / С.О.Пометун // Системні дослідження та інформаційні технології. – 2008, №2, С. 29–40.

Надійшла 14.01.2016 р. Рецензент: д.т.н., проф. Барабаш О.В.