УДК 621.394/681.325

ВИКОРИСТАННЯ СПЕЦІАЛЬНИХ ФУНКЦІЙ ДЛЯ ПРОГРАМНИХ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Танцюра Д.В., Зіньковський Ю.Ф.

Проведено порівняльний аналіз генераторів псевдовипадкових послідовностей на основі однобічної функції з лазівкою. Приведені рекомендації щодо підвищення стійкості та періоду генераторів.

Вступ. Постановка задачі

Програмні та програмно-апаратні засоби генерації псевдовипадкових послідовностей (ПВП) є головними елементами будь якої захищеної комп'ютерної системи, незалежно від її складності та призначення. Серед задач, для рішення яких використовуються генератори ПВП: технічне діагностування, оперативний контроль виконання програм і процесів, моделювання, завадостійке стохастичне кодування, забезпечення безпеки систем (шифрування, програмні системи безпеки, електронний цифровий підпис[1]). Саме від властивостей генераторів ПВП, особливо за необхідності забезпечити надійну стійку роботу програмної системи при наявності випадкових та спланованих деструктивних впливів, в значній мірі залежить надійність процесів накопичення, обробки, зберігання та передавання інформації. До програмних засобів генерації ПВП висувають жорсткі вимоги, в першу чергу за такими параметрами, як непередбачуваність, безпека реалізації, статистичні та періодичні властивості.

У зв'язку з тим, що побудова ідеальних стохастичних пристроїв (пристрої на основі "натуральних" випадкових процесі, наприклад білого радіошуму) викликає певні труднощі, на практиці використовують ПВП чисел, які можна побудувати за допомогою арифметичних алгоритмів (генераторів). Апаратні генератори будуються на окремих мікросхемах з власними функціями, а програмні – на програмному коді який виконується стандартизованими функціям. Найпоширенішими варіантами є програмні генератори, що обчислюють ПВП як складну функцію від даного часу і (або) числа, яке вводить користувач. Найбільш поширені випадкові числа, що рівномірно розподілені на відрізку [0;1], та рівноймовірні двійкові випадкові знаки α_n з умови $P\{\alpha_n = 0\} = P\{\alpha_n = 1\} = 1/2$ (тоді випадковою послідовністю бітів довжини n буде випадкова величина, яка набуває кожне значення із значень $\{0;1\}^n$ з однаковою ймовірністю $(1/2)^n$).

Однобічна функція з лазівкою дає змогу побудувати криптографічні стійкі генератори двійкових ПВП. Такі генератори базуються на тих же важкорозв'язуваних задачах, що й криптографія з відкритим ключем - факторизація складеного числа, добування коренів за модулем, дискретне логарифмування. Наприклад, генератор *BBS* (генератор квадратичних лиш-

ків) - простий та ефективний генератор, що використовує складність факторизації великих чисел.

Принципи побудови і результати дослідження генератора

Можна використати наступні принципи побудови генератора [1,2]:

1. Навмання вибираються прості числа p і q (великі прості числа, приблизно однакового розміру) з властивістю $p \equiv 3(\mathrm{mod}\,4)$, $q \equiv 3(\mathrm{mod}\,4)$ та обчислюється ціле число Блюма ($p \equiv q \equiv 3(\mathrm{mod}\,4)$) $n = p \cdot q$, ці числа p і q зберігаються у таємниці. Стійкість генератора гарантується при умові, що числа p і q прості (хоча на практиці задовольняються і квазі-простими числами): $p\%k \neq 0$, $k = 1 \cdots p$ (1), де % - функція залишку від ділення. Тому необхідно перевіряти їх "на простоту", або заздалегідь вибирати прості числа з відомої множини. Перевірка "на простоту" потребує від 2 до $2p$ операцій.

2. Випадково вибирається з мультиплікативної групи лишків Z_n^* інше ціле число x, взаємно просте з числом n: $(x,n)=1$.

3. Обчислюється число $x_0 \equiv x^2 (\mathrm{mod}\, n)$ як початкове значення генератора.

4. За законом $x_i \equiv x_{i-1}^2 (\mathrm{mod}\, n)$ утворюється послідовність чисел x_i.

5. Шуканою двійковою ПВП $b_1 b_2 \cdots b_m$ ($BBS_{n,m}(x_0)$) буде послідовність молодших бітів чисел x_i, тобто $b_i \equiv x_{i-1} (\mathrm{mod}\, 2)$, $i = \overline{1;m}$.

Стійкість генератора визначається фактом створення обчислювальної системи або ж знаходженням певного алгоритму, за допомогою яких несанкціоновано можна отримати вірне значення початкового стану генератора. Даний генератор можна зламати методом "брутального зламу" або знайшовши прості числа p і q. В першому випадку хакеру відомо число n та алгоритм генератора, тому можна визначити кількість необхідних операцій на кожному кроці.

1. Спочатку визначається кількість операцій для знаходження одного можливого числа x: число x повинно бути взаємно простим з числом n, тому використовуючи алгоритм Евкліда для знаходження найбільшого спільного дільника двох чисел (x,n) $n = xq_0 + r_1$, $x = r_1 q_1 + r_2$, $r_1 = r_2 q_2 + r_3$, $\cdots$, $r_{n-1} = r_n q_n$, r - спільні дільники чисел (x,n). Найбільший спільний дільник чисел (x,n), дорівнює r_n, останньому не нульовому члену цієї послідовності.

Отже, для знаходження одного можливого значення x - необхідно виконати наступні дії: $r_1 = n\%x$, $r_2 = x\%r_1$, $if\,(r_2 = 1) \rightarrow$ *при цьому ключ можливий,* $if\,(r_2 != 1) \rightarrow$ *варіант ключа відкидається*. Отже на цю операцію витрачається 3 дії (за даним алгоритмом).

2. Знаходження можливого початкового стану генератора $x_0 = n + x^2 \% n$. При цьому необхідно виконати щонайменше 3 дії.

3. Для знаходження одного значення можливого генератора

$b_i = \left[x_{i-1}^2 (\mathrm{mod} n)\right](\mathrm{mod}2)$, $i = \overline{1;m}$, де $x_{i-1}^2 (\mathrm{mod}\, n)$ виконується $x_i = (i+1)n + x_{i-1}^2 \% n$. Тобто необхідно 6 дій для знаходження одного значення.

Загальна формула кількості операцій на визначення можливого ключа для певної двійкової ПВП: $L = 3n + 3(1 \cdots n/\ln n)(1+6y)$, де L - загальна необхідна кількість операцій, $(1 \cdots n/\ln n)$– кількість можливих варіантів початкового стану генератора, y – кількість бітів послідовності.

У випадку коли необхідна послідовність має кількість 1024 біт, то в перевіреному програмному варіанті n теоретично може досягати величини від $2^{63} \cdot 2^{63} = 2^{126}$ до $2^{64} \cdot 2^{64} = 2^{128}$. Тобто L в даному випадку буде становити від $\sim 2{,}55 \cdot 10^{38}$ до $7{,}17 \cdot 10^{40}$ операцій.

Визначити i-й біт ПВП без завчасного обчислення i-1 попередніх бітів в звичайному генераторі Блюма можливо за умови, що відоме розкладання модуля на множники: $x^{\varphi(n)} \equiv 1(\mathrm{mod}\, n) \Rightarrow x^{(p-1)\cdot(q-1)} \equiv 1(\mathrm{mod}\, n)$. Звідси $x_i \equiv x_0^{2^i (\mathrm{mod}(p-1)\cdot(q-1))} (\mathrm{mod}\, n)$. Тому за загальним методом факторизації (методом пробного ділення) числа n: $n \% g = 0$, $g = 2,3,5,7 \cdots \le \sqrt{n}$ (алгоритм факторизації) для визначення простих чисел p і q**.** Перше знайдене g, яке задовольняє попередній умові, і буде одним з них (p або q), тоді друге легко визначається. Тому при відомій таблиці простих чисел до $\sqrt{n}$ необхідно $\sqrt{n}/\ln\sqrt{n} + 1$ операцій, в іншому разі – $(\sqrt{n} - 1) \cdot 2 \cdot (\sqrt{n} - 2 + 1)$, а відтак $2{,}11 \cdot 10^{17}$ та $6{,}8 \cdot 10^{38}$операцій відповідно.

Відомий генератор типу *RSA* [2], який будується за допомогою іншої однобічної функції, що є основою криптосистеми *RSA*:

$$x_{i+1} \equiv x_i^e (\mathrm{mod}\, n),\ (e, (p-1)\cdot(q-1)) = 1,\ n = p \cdot q,\ x_0 \prec n;$$

де p і q – великі прості числа (закритий ключ), x_0 - випадкове початкове число. Його стійкість базується на використанні складності факторизації великих чисел. Тому алгоритм факторизації теж підходить для зламу, хоча - недостатній.

Можна визначити кількість операцій для "брутального зламу" програмної реалізації цього генератора.

Так як числа p і q – невідомі, то спочатку необхідно знайти їх. Тоді можна знайти значення $(p-1)(q-1)$. Далі за допомогою умови $(e, (p-1)\cdot(q-1)) = 1$ знаходиться ключ e.

1. Для знаходження одного можливого значення e необхідно виконати наступні дії: $r_1 = (p-1)(q-1)\%e$, $r_2 = e\%r_1$, $if\,(r_2 = 1) \rightarrow$ *при цьому ключ можливий* $if\,(r_2 != 1) \rightarrow$ *варіант ключа відкидається.*

Отже на цю операцію витратиться від 3 до $3(p-1)(q-1)$ дій.

2. У зв'язку з тим, що $x_0 \prec n$, x_0 – випадкове початкове значення генератора, теж невідоме (секретний елемент), але для надійності лежить в межах $p...n$, то можна визначити приблизну кількість операцій необхідних для зламу програмної реалізації цього генератора:

$$L = 2{,}11 \cdot 10^{17} ... 6{,}8 \cdot 10^{38} + 3 ... 3(p-1)(q-1) + (6y)(n-p),$$

тут L - загальна необхідна кількість операцій, y – кількість бітів послідовності. Тобто при $y = 1024$, L буде в межах $5{,}22 \cdot 10^{41}$ до $5{,}24 \cdot 10^{41}$, але існують алгоритми, які швидше (потребують меншої кількості операцій) можуть визначити параметри генератора.

Для генерації (і для зламу) генераторів необхідне вирішення проблеми генерації великих простих чисел. Також для більшої непередбачуваності, унеможливлення відтворення та підвищення надійності програмні генератори комбінують з апаратними, точніше початкові параметри програмних генераторів формують з "натуральних" випадкових процесів: параметри вводить сам користувач, або вони генеруються з часу, встановленому у системі, або вибираються з сигналу звукової, відео карти комп'ютера, або з часових інтервалів між натисканням клавіш користувачем. Тому можна запропонувати наступний генератор, який в деякій мірі можна назвати програмно-апаратним, це варіант генератора Блюма, але зі своїми особливостями. В ньому число p – генерується з числа, яке вводить користувач. Число q – генерується з значень дати і часу системи за допомогою хеш-функції. У випадку, коли не виконується принцип (1), вибирається найближче до генерованого просте число: $p = p - 1$, $p\%k \neq 0$, $k = 1 \cdots p$. Для визначення початкового стану генератора використовується випадкове (зі стандартного простого системного генератора) число k – ціле ($k \prec 10$) і не нуль: $x_0 = kn + x^2 \% n$. Тобто кожний біт ПВП генерується генератором з новим початковим станом, що у багато разів ускладнює "брутальний злам", але потребує глибоких статистичний досліджень (тестів на випадковість та повторюваність). Кожне значення ПВП генерується новим (тим самим, але з новим початковим значенням) генератором, тому можна стверджувати про непередбачувано величезний (в 10…10! разів більший за звичайний генератор Блюма) період розглянутого генератора. Навіть при тих самих початкових значеннях (стану) генератора, відтворена ПВП буде відрізнятися від передбаченої.

Запропонований генератор можна зламати методом "брутального зламу", але знайшовши навіть прості числа p і q, його злам, як генераторів *RSA* та простого генератора Блюма, буде ускладнений, бо можна знайти тільки один біт, а не всю послідовність ПВП**.** Можна визначити кількість необхідних операцій на кожному кроці за умови, що відомо число n та алгоритм генератора (найгірший можливий випадок).

1. Визначення кількості операцій для знаходження одного можливого числа *x*, виконується, як і раніше, за допомогою алгоритму Евкліда. Отже, для знаходження одного можливого значення *x* - необхідно виконати 3 дії.

2. Для знаходження можливого початкового стану генератора $x_0 = kn + x^2 \% n$, k – ціле і не нуль. Необхідно виконати щонайменше 4 дії.

3. Для знаходження одного значення можливого генератора, так само як і в попередньому випадку, необхідно 6 дій для одного значення.

Загальна формула кількості операцій на визначення можливого ключа для певної двійкової ПВП буде: $L = 3n + 4(1 \cdots n/\ln n)k(1+6y)$, де L - загальна необхідна кількість операцій, $(1 \cdots n/\ln n)$ – кількість можливих варіантів початкового стану генератора, *y* – кількість бітів послідовності. При генерації 1024 бітної послідовності, *n* теоретично може досягати значень від $2^{63} \cdot 2^{63} = 2^{126}$ до $2^{64} \cdot 2^{64} = 2^{128}$. Тобто L в даному випадку, при $k = 7$, буде становити від $\succ 2{,}55 \cdot 10^{38}$ до $6{,}61 \cdot 10^{41}$ операцій.

Висновки

Надійність генераторів ПВП базується на випадковості та секретності початкових параметрів, тому доцільно для підвищення надійності та стійкості – використовувати не програмні, а програмно-апаратні генератори, швидкодія яких незначно відстає від програмних. Можна рекомендувати використовувати алгоритм генераторів з непостійним періодом, як це запропоновано в даній статті (модифікований генератор Блюма), але достатнім для необхідної ПВП. Обов'язковим є дослідження генераторів за допомогою статистичних тестів, бо, навіть, складний генератор інколи можна легко передбачити неочікуваними простими функціями. Рекомендувати для криптографічних цілей чисто апаратні генератори недоцільно, оскільки вони обмежені жорсткими вимогами до вбудованої пам'яті, швидкодії та площі (кількості комірок), що ускладнює забезпечення та підвищення криптостійкості.

Література

1.Танцюра Д.В., Зіньковський Ю.Ф. Надійність захисту інформації системи електронного цифрового підпису//Вісник НТУУ «КПІ» Серія – Радіотехніка. Радіоапаратобудування.-2007.-№34, стор.156-163.

2.Математичні основи криптографії: Навч. посібник / Кузнєцов Г.В. та ін. – Дніпропетровськ: Національний гірничий університет, 2004. – Ч. 1. – 391 с.

3.Тесты *DIEHARD* : http://stat.fsu.edu/pub/diehard/

Ключові слова: криптографія, генератор, псевдовипадкової послідовності

Танцюра Д.В., Зиньковский Ю.Ф.

Использование специальных функций для программных генераторов псевдослучайных последовательностей

Проведен сравнительный анализ генераторов псевдослучайных последовательностей на основе функции с уловкой. Приведены рекомендации по повышению стойкости и периода генераторов.

Tantsyura D.V., Zinkovskiy Y.F.

The using of the special function for the program generators pseudorandom sequences

It is organize benchmark analysis generator pseudorandom sequences on base of the unilateral function with trick. Recommendations on increasing of stability and period of generator are brought