Капуста А.О.

Телющенко В.А.

Національний Авіаційний Університет, м.Київ, Україна

Методи створення генераторів псевдовипадкових послідовностей

Генерування випадкових послідовностей великої довжини є однією з важливих проблем класичної криптографії. Для вирішення цієї проблеми широко використовуються генератори псевдовипадкових послідовностей.

Генератор псевдовипадкових послідовностей повинен бути ефективним. Це означає, що він повинен робити послідовності великої довжини за максимально короткий час. Така вимога особливо важливо для систем, що працюють в режимі реального часу. Крім того, генератори псевдовипадкових послідовностей, що застосовуються в задачах криптографії, повинні бути стійкі до різних атак і нестандартних ситуацій.[1-2]

Існує величезна кількість генераторів випадкових послідовностей, кожен з яких має свої плюси і мінуси. Найбільш широковідомим є лінійний конгруентний метод.

Лінійний конгруентний метод поширений метод для генерації псевдовипадкових послідовностей, що не володіє криптографічного стійкістю.[3]

Недоліком лінійно конгруентного методу в плані його використання для створення потокових шифрів є передбачуваність вихідних послідовностей. Ефективні атаки на лінійно конгруентний генератор були запропоновані Joan Boyar, їй належать методи атак на квадратичні і кубічні генератори.

Інші дослідники узагальнили результати робіт Boyar на випадок загального полиномиального конгруентного генератора. Stern і Boyar показали, як зламати конгруентний генератор, навіть якщо відома не вся послідовність.[4-5]

Перевагою лінійних конгруентних генераторів псевдовипадкових чисел є їх простота і висока швидкість отримання псевдовипадкових значень. Лінійні

конгруентні генератори знаходять застосування при вирішенні задач моделювання і математичної статистики, проте в криптографічних цілях їх не можна рекомендувати до використання, так як фахівці з криптоаналізу навчилися відновлювати всю послідовність псевдовипадкової послідовності за кількома значеннями.[6]

У більшості мов програмування саме цей метод використовується в стандартній функції отримання псевдовипадкових послідовностей. Вибирається 4 числа:

- Модуль m (m> 0);
- Множник a (0 <= a <m);
- Приріст c (0 <= c <m);
- Початкове значення X0 (0 <= X0 <m)

Послідовність виходить з використання наступної рекурентної формули: $X_{n+1} = (a * X_n + c) mod\ m$

Цей метод дає дійсно хороші псевдовипадкові послідовності, але, якщо взяти числа m, a, c довільно, то результат нас швидше за все розчарує. При m = 7, X_0= 1, a = 2, c = 4 вийде наступна послідовність: 1,6,2,1,6,2,1, ...

Очевидно, що ця послідовність не зовсім підходить під визначення випадкової. Проте, цей провал дозволив нам зробити два важливих висновки:

1. Числа m, a, c, X_0 не повинні бути випадковими;
2. Лінійний конгруентний метод дає нам повторювані послідовності.[7]

Насправді будь-яка функція, що відображає кінцеве безліч X в X, буде давати циклічно повторювані значення. Тобто наше завдання полягає в тому, щоб максимально подовжити унікальну частину послідовності (до речі, очевидно, що довжина унікальної частини не може бути більше m).

Не вдаючись в подробиці доказів, скажімо, що період послідовності буде дорівнювати m тільки при виконанні наступних трьох умов:

1. Числа c і m взаємно прості;

2. a-1 кратно p для кожного простого p, що є дільником m;

3. Якщо m кратно 4, то і a-1 повинна бути кратним 4.

Можна сказати, що послідовності, одержувані з його допомогою, хоч і є в достатньому сенсі випадковими, проте не є криптографічно стійкими. Оскільки знаючи 4 поспіль числа, криптоаналітик може скласти систему рівнянь, з яких можна знайти a, c, m. [8]

Отже, лінійні конгруентнні генератори мають простий алгоритм і високу швидкість роботи. Однак використовувати в криптографії не рекомендується, так як їх легко «зламати». Тобто, маючи послідовність чисел, отриману таким генератором, опонент може відновити параметри генератора, витративши мінімум зусиль.

Вирішити цю проблему можна за допомогою програмного модуля. Для здійснення генерації чисел даним методом, нам необхідно підібрати 4 числа. Варто зауважити, що в даному методі багато що залежить від підбору параметрів.

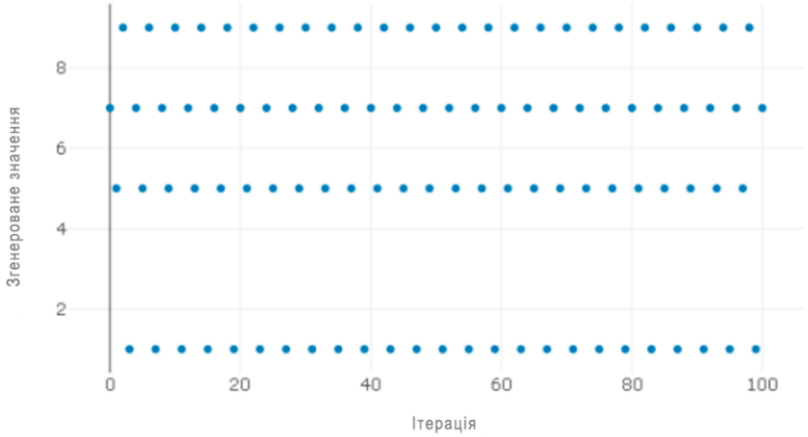
Наприклад, для наступного набору:

$$\begin{cases} X_0 = 7 \\ a = 8 \\ c = 9 \\ m = 10 \end{cases}$$

ми отримаємо коротку повторювану послідовність

7, 5, 9, 1, 7, 5, 9, 1 ...

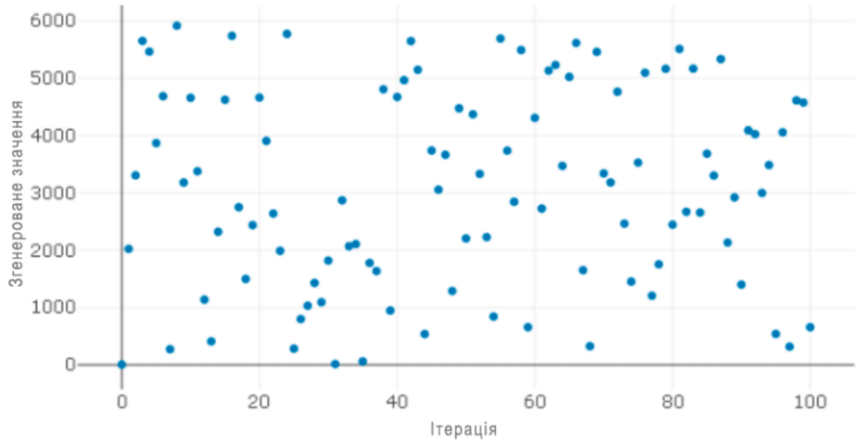
яка зовсім не виглядає випадковою:



Але варто змінити параметри на наступні:

$$\begin{cases} X_0 = 7 \\ a = 106 \\ c = 1283 \\ m = 6072 \end{cases}$$

І віра в те, що алгоритм дійсно здатний генерувати псевдовипадкові числа міцнішає:



Отже, якщо правильно підбирати вхідні параметри, використовуючи прийняті коефіцієнти, якість послідовностей, що генеруються буде відповідати сучасним вимогам.

Література

1. https://ru.abcdef.wiki/wiki/Linear_congruential_generator
2. https://books.ifmo.ru/file/pdf/2474.pdf
3. https://cmcmsu.info/1course/random.generators.algs.htm
4. https://studopedia.ru/10_213782_generatori-psevdosluchaynih-posledovatelnostey.html
5. https://pandia.ru/text/77/481/8848.php
6. https://intuit.ru/studies/courses/691/547/lecture/12383?page=2
7. https://habr.com/ru/post/132217/
8. https://math.stackexchange.com/questions/3838018/limit-to-values-of-a-linear-congruential-generator