МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

Факультет математики, фізики та інформаційних технологій

Кафедра комп'ютерної алгебри та дискретної математики

Дипломна робота

на здобуття ступеня вищої освіти «бакалавр»

на тему

«Однонаправлені функції в криптографії»

«Unidirectional functions in cryptography»

Виконав: студент денної форми навчання

спеціальності 123 – Комп'ютерна інженерія

Авагян Ваган Ваагнович

Керівник: Варбанець П. Д.

Рецензент: Бєлозьоров В. Г.

Рекомендовано до захисту:

Протокол засідання кафедри

№ ___від «_____»_____2021 р.

Завідувач кафедри

_____ Варбанець П.Д.

Захищено на засіданні ЕК №___

протокол №__ від «__»__2021 р.

Оцінка / /

Голова ЕК

_____ Казакова Н. Ф.

Одеса – 2021

АНОТАЦІЯ

У дипломній роботі розробляється тема "Однонаправлені функції в криптографії".

Метою роботи є побудова класу односпрямованої функції за допомогою конгруетного генератора та вивчення властивості непередбачуваності генераторів псевдовипадкових чисел.

АННОТАЦИЯ

В дипломной работе разрабатывается тема "Однонаправленные функции в криптографии".

Целью работы является построение класса однонаправленной функции с помощью конгруетного генератора и изучение свойства непредсказуемости генераторов псевдослучайных чисел.

ANNOTATION

In the thesis the topic "Unidirectional functions in cryptography" is being developed.

The aim of the work is to build a class of a one-way function using a congruent generator and to study the unpredictability property of pseudo-random number generators.

ЗМІСТ

ВСТУП

Класична симетрична криптографічна система характерна тим, що знання способу шифрування та розшифрування еквівалентні. Але у нових криптосистемах справа інша: знаючи ключ шифрування, ми не можемо розшифрувати секретне повідомлення уникнувши надзвичайно довгих обчислень. У криптосистемі з відкритим ключем маємо ключ шифрування E (m)=c (тобто f(m)=c), де m-вихідне повідомлення, а c-зашифроване повідомлення. Ми не можемо (за короткий час) по образу c відновити m, тобто ми не знаємо функцію f^{-1}:c->m. Отже, ми не можемо знайти m без деякої додаткової інформації. Функцію f у такому випадку називають функцією з замком. Дуже близким до цієї функції є однонаправлена (одностороння) функція. Зазвичай паролі шифрування та розшифрування розглядаються як цілі числа за великим модулем M, а односпрямоване відображення задається многочленом великого ступіню mod M: Zm$\rightarrow^{f}$Zm. Многочлен f можна брати як той, що містить обмежене число доданків. Але оскільки ступінь f(x)-велика, а місце розташування доданків секретно, то відновити значення x0 за значенням f(x0) практично неможливо.

У даній дипломній роботі ми використовуємо цю ідею побудови односпрямованої функції, застосовуючи конгруетні генератори псевдовипадкових чисел.

Існує два типа генераторів псевдовипадкових чисел: лінійний конгруентний генератор і нелінійний конгруентний генератор. У свою чергу, нелінійні конгруенціальні генератори діляться на: інверсний, квадратичний тощо. У даній дипломній роботі більш детально буде вивчений інверсний конгруентний генератор псевдовипадкових чисел.

ПОСТАНОВКА ЗАДАЧІ

Основним завданням дипломної роботи є побудова класу односпрямованої функції за допомогою конгруетного генератора та вивчення властивості непередбачуваності генераторів псевдовипадкових чисел. Метою досліджень була побудова послідовностей псевдовипадкових чисел за допомогою інверсного конгруентного генератора, а так само вивчення впливу параметрів генератора a, b, x0, m, r на характер непередбачуваності псевдовипадкової послідовності шляхом проведення чисельних експериментів. Для цього потрібно розглянути генератор з «малими» та «великими» значеннями параметра a. Аналогічне дослідження провести для значення x0. Крім того, згенерувати змішані послідовності, породжені інверсним генератором.

Відомо, що серіальні тести для випадкових послідовностей можуть бути використані для оцінки непередбачуваності елементів таких послідовностей. Подібним тестом може служити рівномірність розподілу двовимірних точок виду (x_0, x_1), (x_2, x_3), … в одиничному квадраті. Розробити програму, що дозволяє будувати генератори даного типу і візуалізувати отримувані послідовності.

1. ЛІНІЙНИЙ КОНГРУЕНТНИЙ МЕТОД

1.1 Загальні відомості

Найкращі з відомих сьогодні датчиків випадкових чисел є окремі випадки такої схеми, запропонованої Д. Лемером [1] в 1949 р. Вибираємо чотири "магічних числа":

1. x_0 початкове значення; $0 < x_0 < m$

2. a множник; $0 < a < m$

3. c прирощення; $0 \leq c < m$

4. m, модуль; $0 < m$

Тоді шукана послідовність випадкових чисел $\{x_n\}$ виходить із співвідношення

$$x_{n+1} = (a * x_n + c) \bmod m \qquad (1)$$

Вона називається лінійной конгруентной послідовністю. Наприклад, при $x_0 = a = c = 7$, $m = 10$ послідовність виглядає так:

7, 6, 9, 0, 7, 6, 9, 0 (2)

Як видно з наведеного прикладу, послідовність не завжди виявляється "випадковою", якщо вибирати x_0, a, c, m довільно. У цьому розділі будуть коротко висвітлені принципи вибору цих значень.

Приклад ілюструє той факт, що конгруентні послідовності завжди "зациклюються", тобто в кінці кінців числа утворюють цикл, який повторюється нескінченне число разів. Ця властивість властива всім послідовностям, які мають загальний вигляд $x_{n+1} = f(x_n) \bmod m$. повторюваний цикл називастся періодом.

Довжина періоду у послідовності (2) дорівнює 4. Реальні послідовності, якими користуються, мають порівняно великий період.

Спеціальної згадки заслуговує окремий випадок c = 0, коли процес вироблення випадкових чисел відбувається трохи швидше. Обмеження c = 0 зменшує довжину періоду послідовності, але при цьому все ще можна отримати відносно великий період. У первісному методі Лемера було прийнято c = 0, хоча автор і згадав можливість використання $c \neq 0$. Ідея отримання більш довгих послідовностей за рахунок узагальнення $c \neq 0$ належить Томпсону [2] і незалежно Ротенбергу [3].

Багатьма авторами терміни мультиплікативний конгруентний метод і змішаний конгруентний метод застосовуються для позначення лінійних конгруентних методів з c = 0 і $c \neq 0$ відповідно.

Позначимо:

$$b = a-1 \tag{3}$$

Можна відразу ж відкинути випадок a = 1, так як при цьому $x_n = (x_0 + n * c) \bmod n$, і очевидно, що послідовність не є випадкова. Варіант a = 0 ще гірше. Отже, для практичних цілей ми можемо припустити, що:

$$a \geq 2,\ b \geq 1 \tag{4}$$

Тепер можна узагальнити співвідношення (1):

$$x_{n+k} = (a^k x_n + (a^k - 1)c/b) \bmod m \tag{5}$$

1.2 Вибір модуля і множника

Для будь-якій послідовності, призначеної для використання в якості джерела випадкових чисел, важливим є великий період. Необхідно, щоб значення m було досить великим, так як довжина періода не може бути більше m. Навіть якщо потрібні тільки випадкові нулі і одиниці, не слід брати m = 2, так як при тому в кращому випадку вийде набір:

0, 1, 0, 1, 0, 1, ...

Так як можливі тільки m різних значень, довжина періоду не може перевищувати m. Чи досяжна максимальна довжина m? Наприклад, при a = c = 1 послідовність зводиться просто до x_{n+1} = (x_n + 1) mod m. Очевидно, її період дорівнює m, проте, її ніяк не можна назвати випадковою. Досліджуємо всі можливі способи вибору a й c, які дають період довжини m. Коли період має довжину m, кожне число від 0 до m - 1 зустрічається за період рівно один раз. Тому вибір x_0 не впливає на довжину періоду.

Теорема 1:

Довжина періоду лінійної конгруентної послідовності дорівнює m тоді і тільки тоді, коли:

1. c і m - взаїмно прості числа;

2. b = a - 1 кратно p для будь-якого простого p, що є дільником m;

3. b кратно 4, якщо m кратно 4.

Ідеї доказу відомі не менше ста років. Перший доказ теореми в даному формулюванні було дано М. Грінбергером [4] для окремого випадку m = 2^e. Достатність умов 1, 2 і 3 в загальному випадку була показана Халлом і Добеллом [5].

Лемма 2:

Нехай p - просте число, а e - позитивне ціле, таке, що $p^e > 2$.

Якщо

$x \equiv 1 (\mathrm{mod}\ p^e)$ та $x \neq 1 (\mathrm{mod}\ p^{e+1})$

То

$x^p \equiv 1 (\mathrm{mod}\ p^{e+1})$ та $x^p \neq 1 (\mathrm{mod}\ p^{e+2})$

1.3 Потенціал

Великий період - це тільки один з необхідних ознак випадковості послідовності. Цілком можливі абсолютно невипадкові послідовності з дуже великим періодом. Це ілюструє наведений вище приклад з a = c = 1. Одна з причин цього пов'язана з концепцією "потенціалу".

Потенціал лінійної конгруентної послідовності з максимальним періодом визначається як найменше ціле s, таке, що

$b^s \equiv 0$ (по модулю m)

(Таке ціле s завжди існує, якщо b кратно будь-якому простому дільнику m)

Ми можемо аналізувати випадковість послідовності, приймаючи $x_0 = 0$, так як нуль обов'язково зустрічається на протязі її періоду. У такому випадку маємо

$x_n = (a^n - 1)\, c / b \bmod m$

і, розклавши $a^n - 1 = (b + 1)^n - 1$ за формулою бінома, знаходимо

$$x_n = c\,(n + \binom{n}{2} b + \ldots + \binom{n}{s} b^{s-1}) \bmod m \qquad (6)$$

Звідси видно, що всі члени з b^s, b^{s+1} і т.д. можна опустити, так як вони кратні m. Виходячи з рівняння (6), розглянемо деякі приватні випадки. Якщо a = 1, потенціал дорівнює 1 і, як ми вже бачили, $x_n \equiv cn$ (по модулю m), так що послідовність явно не випадкова. Якщо потенціал дорівнює 2, маємо $x_n \equiv cn + cb\binom{n}{2} \bmod m$, і знову послідовність не можна вважати випадковою. Дійсно,

$x_{n+1} - x_n \equiv c + cbn$,

так що різниця між сусідніми випадковими числами виражається дуже простою залежністю від n. Нехай d = cb mod m, тоді точка (x_n, x_{n+1}, x_{n+2}) завжди лежить на одній з чотирьох площин в тривимірному просторі:

x – 2y + z = d + m

x – 2y + z = d

x -2y + z = d – m

x -2y + z = d – 2m

Якщо потенціал дорівнює 3, послідовність виглядає кілька більш випадковою. Але x_n, x_{n+1}, x_{n+2} все ще пов'язані сильною залежністю. Різниці x_{n+1} - x_n утворюють послідовність з потенціалом 2, і тести показують, що послідовності з потенціалом 3 все ще недостатньо гарні. Повідомлялося, що прийнятні результати можуть бути отримані для значення потенціалу, рівного 4 і вище, але це було оскаржено багатьма авторами. Мабуть, для досить випадкових значень потрібен потенціал, рівний, щонайменше, 5.

Як приклад розглянемо наступну послідовність: m = 2^{35} та a = 2^k + 1. Тоді b = 2^k, так що величини b^2 = 2^{2k} кратні m, коли k ≥ 18: потенціал дорівнює 2. Якщо k = 17, 16, …, 12, то потенціал дорівнює 3 і значення потенціалу 4 досягається для k = 11, 10, 9. Таким чином, з точки зору потенціалу множителі прийнятні при k ≤ 8. Це означає, що a ≤ 257, але малих множників теж слід уникати. Отже, всі множники виду 2^k + 1 при m = 2^{35} виключені.

На жаль, велике значення потенціалу (наприклад ≥ 5) ще не гарантує гарну випадковість. В даний час надійним тестом на випадковість довільної послідовності дійсних чисел $\{y_k\}$, $0 \le y_k \le 1$, k = 0, 1, … є спектральний тест. Його суть зводиться до того, щоб довести, що точки $M_{n,k}$ = (y_n, y_{n+1}, …, y_{n+k-1}) рівномірно розподілені в k-вимірному одиничному гіперкубі $[0,1)^k$. Це призводить до необхідності вивчення тригонометричних сум виду:

$S = \sum_{n=1}^{N} e^{2\pi(h_1 y_n + \cdots + h_k y_{n+k})}$, при $((h_1, \ldots, h_k) \neq (0, \ldots, 0))$

Але отримати нетривіальну оцінку суми S для лінійної конгруенціальної послідовності $\{y_k\}$, де $y_k = \frac{x_k}{m}$, важко. Значить, лінійні конгруенціальні послідовності (1), взагалі кажучи, не задовольняють властивості непередбачуваності, а тому непридатні для криптографічного застосування. Тому в наступному параграфі ми розглянемо спеціальний випадок нелінійних конгруенціальних послідовностей.

2. ІНВЕРСНИЙ КОНГРУЕНТНИЙ МЕТОД

2.1 Загальні відомості

Лінійні генератори вимагають малої кількості обчислень для отримання чергового елемента послідовності і можуть мати максимальний період (рівний модулю обчислень). Генеровані точки мають рівномірний розподіл. Однак досить важко домогтися прийнятної випадковості - у більшості випадків послідовність псевдовипадкових чисел, що була згенеровувана лінійним генератором має чітко виражені закономірності. Серед лінійних генераторів є такі, які цілком відповідають вимозі випадковості, проте їх побудова досить затратна. Отже, такі генератори цілком можна застосувати в задачах статистики, в силу того, що виконується умова рівномірності, проте не можуть бути використані в криптографії, так як ключі шифрування, отримані за допомогою лінійних генераторів, будуть уразливі до деяких видів атак. Тому були введені в розгляд нелінійні конгруентні генератори.

Нехай p - просте, a й b - елементи кінцевого поля GF(p) = Z/Zp. Послідовність X (x_0; a, b) = x_0, x_1 ,… , яка визначається співвідношенням:

x_{n+1} = (a * x_n^{-1} + b) mod p якщо $x_n \neq 0$

x_{n+1} = b якщо $x_n = 0$

називається інверсною конгруентною послідовністю.

Ейченауер і Лен [7] використовували співвідношення для генерації псевдовипадкових послідовностей. Нідеррайтер [8] узагальнив цей метод для довільного кінцевого поля. Проблема довжини періоду такої рекурсівної инверсної послідовності була детально досліджена в роботах Чоу [9], Ейченауера і Лена [7], Флахіва і Нідеррайтера [10].

Для перехода до генерації псевдовипадкових чисел з довільним позитивним модулем, (7) має бути змінено в такий спосіб:

x_{n+1} = (a * x_n^{-1} + b) mod m при НСД (x_n, m) = 1 (8)

Таким чином кожен терм X (x_0; a, b) повинен бути взаємно простим з модулем m. Якщо m = $P_1^{r_1}$... $P_t^{r_1}$ де t ≥ 2 та $P_1 \ldots P_t$ - попарно різні прості числа, то довжина періоду послідовності X (x_0; a, b) з модулем m дорівнюватиме найменшому спільному кратному довжин періодів послідовностей X (x_0; a, b) з модулем $P_i^{r_i}$, 1 ≤ i ≤ t. Тому для вивчення довжини періоду послідовності X (x_0; a, b) з модулем m достатньо вивчити довжину періоду послідовності X (x_0; a, b) з модулем p^k, де p – просте. Ейченауер, Лен та Топузоглу [11] розглянули умови отримання максимальної довжини періоду для модуля 2^k. Оскільки послідовність X (x_0; a, b) по модулю p - простого дільника m замість m не містить 0 по модулю p, ця послідовність не може досягати максимального періоду. Важливо вивчити всі можливі довжини періодів рекурсії (7) над кінцевим полем GF(p). Ейченауер і Лен [7] отримали деякі результати щодо довжин періодів X (x_0; a, b) з простим модулем. Чоу [12 | узагальнив це для кінцевого поля і отримав всі можливі довжини періодів послідовності X (x_0; a, b).

В силу теореми Ейлера маємо:

x * $x^{\phi(m)-1}$ = 1 (mod m)

для кожного x при НОД (x, m) = 1. Тому генератор (8) можна замінити генератором

x_{n+1} = (a$x_n^{\phi(m)-1}$ + b) mod m для усіх n ≥ 0 (9)

Де ф(x) - функція Ейлера Це співвідношення еквівалентно співвідношенню (7) якщо m є простим числом і еквівалентним співвідношенню (8) коли кожен терм послідовності X (x_0; a, b) по модулю m взаємно простий з m. При цьому рекурсія (9) допускає будь-який терм x_n, навіть не взаємно простий з m. Генератор (9) будемо називати узагальненим інверсним генератором.

Займемося визначенням періоду r для інверсних конгруентних послідовностей (8) (відповідно (9)).

Визначення:

Порядком многочлена f(x) над кінцевим полем GF(p) називається найменше натуральне k, для якого виконується: x^k- 1: f(x) в кільці GF(p)[x].

Лемма 1:

Нехай p - довільне просте число, x_0, a и b - елементи деякого кінцевого поля GF(p). Позначимо через X (x_0; a, b) послідовність, отриману зі співвідношення (7), f(x) = x^2 – bx - a, o(mf) - порядок многочлена mf(x) = x^2 + (b^2/ a + 2) * x +1 при a ≠ 0 і L(x_0; a, b; p) - довжина періода X (x_0; a, b). Тоді вірні наступні твердження:

A)Якщо a = 0 то x_n = b для усіх n ≥ 1, отже L(x_0; a, b; p) = 1.

B)Якщо a≠0 та b=0, x_0=0 то x_n = 0 для усіх n ≥ 1, отже L(x_0; a, b; p) = 1.

C)Якщо a*x_0≠0,a=x_0^2 і b=0 то x_n=x_0 для усіх n ≥ 1,отже L(x_0; a, b; p) = 1.

D)Якщо a*x_0≠0,a≠x_0^2 і b=0 то x_n+2=x_n для усіх n≥1,отже L(x_0;a,b;p) = 2.

E)Нехай f(x)=(x − α)^2 для деяких α Є GF(p) (або,теж саме b^2+4a=0).Тоді

1)L(α;a,b;GF(p))=1

2)Якщо x_0≠ α то X (x_0;a,b) містить 0 і L(x_0;a,b;p) = p-1

F)Нехай f(x)=(x-α)*(x-β) для деяких α,β Є GF(p^2),α≠β. Тоді

1)Якщо f(x_0)=0 то L(x_0; a, b; p) = 1

2)Якщо p≠2 і o(mf)- парне то X(b/2;a,b) містить 0 і L(b/2;a,b)=o(mf)-1

3)Якщо p≠2 і o(mf)- непарне то X(b/2;a,b) не містить 0 і L(b/2;a,b)=o(mf)

4)Якщо f(x_0)≠0, x_0≠b/2,p≠2 і o(mf)-порядок многочлена Mf(x)=x^2-(2+(b^2+4a)/f(x_0))x+1 над кінцевим полем GF(p) ділить o(mf),то X(x_0;a,b) містить 0 і L(x_0;a,b;p) = o(mf)-1

5)Якщо $f(x_0)\neq 0$, $x_0\neq b/2$,$p\neq 2$ і o(mf)-не ділить o(mf), то $X(x_0;a,b)$ не містить 0 і $L(x_0;a,b;p) = o(mf)$

Ісльозуя цю лему, Чоу [6] розглянув ше можливі довжини періодів послідовності $X(x_0; a,b)$ по модудю m, згенеровані співвідношеннями (7) і (8).

2.2 Інверсна конгруенціальна рекурсія

Нехай $m \geq 4$ - фіксоване складене ціле і нехай $m = P_1^{r_1}\ldots P_t^{r_t}$, де $t \geq 2$ і $p_1\ldots p_t$ - попарно різні прості числа, $r_1\ldots r_t$ - цілі позитивні. $X(x_0; a,b)$ - послідовність, певна співвідношенням (8) для деяких a, b і x_0. Як уже згадувалося вище, кожен терм цієї послідовності повинен бути взаімно простий з m і довжина періоду $L(x_0; a, b; m)$ послідовності $X(x_0; a,b)$ дорівнює найменшого спільного кратного довжин періодів $L(x_0; a, b; P_i^{r_i})$ послідовностей $X(x_0; a,b)$ по модулю $P_i^{r_i}$, $1 \leq i \leq t$.

Лемма 2:

Нехай p- просте, k, a, b, x_0 – цілі. $k \geq 2$. Нехай $f(x) = x^2 - bx - a$. Тоді a, b, x_0 можуть скласти нескінченну послідовність $X(x_0; a,b)$ по модулю p^k, описувану співвідношенням (8) в тому і тільки в тому випадку, якщо:

A) $a = 0 \bmod p$ та НСД $(bx_0, p) = 1$

B) НСД $(ax_0, p) = 1$ та $b = 0 \bmod p$

C) НСД $(abx_0, p) = 1$, $b = 2x_0 \bmod p$ та $a = -x_0^2 \bmod p$

D) НСД $(abx_0(b^2 + 4a), p) = 1$ та $ax^2 - x_0 b - a = 0 \bmod p$

E) $p \neq 2$, НСД $(ab(b^2 + 4a), p) = 1$, $x_0 = b/2 \bmod p$ і порядок o(mf) многочлена $mf(x) = x^2 + (b^2 / a + 2)x + 1$ над кінцевим полем GF(p)[x] є парним числом.

F) НСД $(abx_0(b^2 + 4a) * f(x_0), p) = 1$, $x_0 \neq b/2 \bmod p$ якщо $p \neq 2$ й порядок o(mf) многочлена $Mf(x) = x^2 - (2 + (b^2 + 4a)/f(x_0)) * x + 1$ над кінцевим полем GF(p)[x] не ділить o(mf).

Наступні дві леми отримані Ейченауер-Херманом і Топузоглу. Вони корисні при описі довжин періодів послідовностей X(x_0; a,b) по модулю p^k при k ≥ 2.

Лемма 3:

Нехай p - просте, k, a, b і x_0 - довільні цілі, k ≥ 2 і НСД (a, p) = 1. Нехай X(x_0; a,b) - послідовність, отримана з рекурсивного співвідношення (8). Нехай λ_{k-1} і λ_k - довжини періодів послідовностей X(x_0; a,b) з модулями p^{k-1} і p^k відповідно. Тоді

A) $\lambda_{k-1} = \lambda_k$ при x$\lambda_{k-1} = x_0 \bmod p^k$

B) λ_k = o(-ax_0^{-2}) λ_{k-1} при x$\lambda_{k-1} \neq x_0 \bmod p^k$, λ_{k-1} = 1 і НСД(a + x_0^2, p) = 1, де o(-ax_0^{-2}) це мультиплікативний порядок многочлена -ax_0^{-2} в GF(p).

C) p$\lambda_{k-1} = \lambda_k$ при x$\lambda_{k-1} \neq x_0 \bmod p^k$ та $\lambda_{k-1} \geq 2$ чи λ_{k-1} = 1 та a = -x_0^{-2} mod p.

Лемма 4:

Нехай p - просте, k, a, b і x_0 - довільні цілі, k ≥ 2 і НСД (a, p) = 1. Нехай X(x_0; a,b) - послідовність, отримана з рекурсивного співвідношення (8). Нехай λ_{k-1} і λ_k - довжини періодів послідовностей X(x_0; a,b) з модулями p^{k-1} і p^k відповідно. Тоді

A) Якщо k ≥ 3 і x$\lambda_{k-1} \neq x_0 \bmod p^k$ тоді x$\lambda_k \neq x_0 \bmod p^{k-1}$

B) Якщо $\lambda_1 \geq 2$ і x$\lambda_1 \neq x_0 \bmod p^2$ тоді x$\lambda_2 \neq x_0 \bmod p^3$

C) Якщо λ_1 = 1, a = -x_0^{-2} mod p, x1 ≠ x_0 mod p^2 і p ≥ 5 тоді x$\lambda_2 \neq x_0 \bmod p^3$

Виходячи з попередніх лем, Чоу [6] довів таку теорему, яка буде прийнята нами за основу при побудові інверсного конгруентного генератора.

Теорема 5:

Нехай p - просте, k, a, b і x_0 - цілі, k ≥ 1. Нехай X(x_0; a,b) - послідовність, отримана з рекурсивного співвідношення (8), f(x) = x^2 – bx - a.

A) Якщо a=0 mod p і НСД (bx_0, p) = 1 тоді довжина періоду X(x_0; a,b;p^k)=1

B) Нехай НСД (ax_0, p)=1, a=x_0^2 mod p і b = 0 mod p. Приймемо b=dp^j при НСД(d, p)=1 при b≠0. Також приймемо f(x_0)=cp^e, НОД (c, p)=1 при f(x_0)≠0

1) Якщо f(x_0)=0 чи k≤e то L(x_0;a,b; p^k)=1

2) Якщо k=e+1 то L(x_0;a,b; p^k)=2

3) Якщо e+1<k і b=0 чи k≤j, то L(x_0;a,b; p^k)=2

4) Якщо k>e+1 і k>j то L(x_0;a,b; p^k)=2$p^{k-\max\{j,e+1\}}$

C) Нехай НСД (ax_0(a-x_0^2),p) = 1 і b = 0 mod p.

1) Якщо b=0, то L(x_0;a,b; p^k)=2

2) Якщо b=dp^j при НСД(d,p)=1 тоді L(x_0;a,b; p^k)=2 при 1≤k≤j і L(x_0;a,b; p^k)=2p^{k-j} при k>j

D) Нехай НСД(ab,p)=1,b=2x_0 mod p і a=-x_0^2 mod p. Якщо f(x_0)≠0 то f(x_0)=cp^e при НСД(c,p)=1

1) Якщо f(x_0)-0 чи e≥2 і k≤e тоді L(x_0;a,b; p^k)=1

2) Якщо k>e≥2 тоді L(x_0;a,b; p^k)= p^{k-e}

3) Якщо p≥5 і e=1 тоді L(x_0;a,b; p^k)= p^{k-1}

4) Нехай p=3 і e=1. Нехай a+b^2=h3^s для деякого цілого h, такогоб що НСД(h,3)=1 при a+b^2≠0. Тоді L(x_0;a,b; 3^k)=3 якщо a+b^2=0 чи 2≤k≤s і L(x_0;a,b; 3^k)= 3^{k-s+1} якщо k≥s+1

E) Нехай НСД (abx_0(b^2+4a),p)=1. Запишемо f(x_0)=cp^e при НСД(c,p)=1 при f(x_0)≠0

1) Якщо f(x_0)=0 чи k≤e тоді L(x_0;a,b; p^k)=1

2) Якщо k>e≥2 тоді L(x_0;a,b; p^k)=o(-ax_0^2)p^{k-e-1}

3) Нехай k>e і ч=o(-ax_0^2) тоді L(x_0;a,b; p^k)= ч якщо $x_ч$=x_0 mod p^k і L(x_0;a,b; p^k)= чp^{k-t+1} де t це найменше число, таке, що $x_ч$≠x_0 mod p^t і 3≤t≤k.

F) Нехай НСД (abx$_0$(b^2+4a)f(x$_0$),p)=1. Покладемо p>2, x$_0$=b/2 mod p і порядок λ=o(mf) многочлена mf=x^2+(b^2/a+2)x+1 в поле GF(p)[x] є непарним числом або порядок o(Mf) многочлена Mf(x)= x^2 - (2+(b^2+4a)/f(x$_0$)) x + 1 в поле GF(p)[x] не поділяє λ. Тоді p непарний і L(x$_0$;a,b; p^k)= λ якщо x$_\lambda$ = x$_0$ mod p^k і L (x, ; a, b; p *) - AP "- де 1 -наіменше ціле, яке задовольняє співвідношення x, + x, mod p 'i 2 <tsk.

Оскільки випадки B) і C) для нас найбільш цікаві, наведемо доказ теореми для цих випадків.

Доказ

B) З визначення тоді коли f (x,)% 3D тоді тільки x - bx, - a - Omod p '. Тому якщо f (x,)% - 0 то L (xo; a, b; p *)% 3D1. Тепер розглянемо випадок, коли f (x,) 2 0. Очевидно, що якщо до S e, то L (x ,; a, b; p ")% 3D 1. Будемо виходити з того, що до> e. тоді x, і x, mod p ". Якщо k -e + 1 ip-непарне, тоді по лемі 3 (B), L (x ,; a, b; p" ") - 2 т. к. o (- ax) - 2. По лемі 3 (C), якщо p -четное і k% 3D e + 1, тоді

Тепер припустимо, що до> e + 1. За оп ределению x, x, tod p * тоді і тільки тоді коли Спростивши це конгруенціальное рівняння, отримаємо x, в X, tod p "тоді і тільки тоді коли ab 0 mod p '. Т.е. якщо Ь 0 або k si. Тогда L (x ,; a, b; B p ') -2 по лемі 3 (A). Якщо до> j. по лемі 3

(C) Із леми 1 (A). леми 2. визначенню 0, очевидно, що L (x ,; a, b; p *) - 2. Розглянемо випадок b 0. Після спрощення отримаємо: x, x, tod p "тоді і тільки тоді коли (a-x,) b-x, b 'mod p". Це виконується в тому і тільки в тому випадку, сслі 1 Sk Sj. тому b dp 'при НСД (a-x ,, p) -1. Тому L (x ,; a, b; p ') -2 при 1 <ksj.

Оскільки (A -x3) ь і x, b 'mod p' ", по лемі 3 (C) L (x,: a, b; p '!) - 2p. Якщо К> j то L (x ,, a , b; p ') -2p * по лемі 3 (C) і леми 4 (A) і (B).

Знову, як і при аналізі «випадковості» розподілу членів лінійної конгруенціальной послідовності, ми повинні розглядати розподіл двійок, трійок і тд. в двовимірному, тривимірному і т.д. одиничному гиперкубі. Дослідження, проведені в роботах Нідеррайтера і Шпарлінскі [14] і Варбанца [15], показують, що нормалізована послідовність (тобто модуль) породжує двовимірні і тривимірні точки, які рівномірно розподілені в двовимірному і, відповідно тривимірному одиничному гиперкубе, а тому відповідна послідовність псевдовипадкових чисел цілком непередбачувана і може застосовуватися в криптографічних додатках.

3. СТАТИСТИЧНІ КРИТЕРІЇ

Як говорилося раніше, основною метою є отримати послідовність, яка поводиться так, ніби вона є випадковою. Ми розглянули, як досягти максимального періоду послідовності, але цей критерій не дасть гарантії, що послідовність буде використовуватися в додатках. Як дізнатися чи буде послідовність випадковою?

Теоретична статистика надає деякі кількісні заходи випадковості. Існує велика кількість різних крітерієв для перевірки того, чи буде послідовність випадковою. Розглянемо найбільш пристосовані до обчислень на комп'ютері критерії.

Якщо критерії T1, T2, …, Tn підтверджують, що послідовність поводиться випадковим чином, це ще не означає, що перевірка Tn+1-го критерію буде успішною. Однак кожна успішна перевірка дає все більше впевненості в випадковості послідовності.

Розрізняють два види критеріїв: емпіричні критерії, при використанні яких комп'ютер маніпулює групами чисел послідовності і обчислює певні статистики, і теоретичні критерії, для яких характеристики послідовності визначаються з допомогою теоретико-числових методів, заснованих на рекурентних правилах, які використовуються для утворення послідовності.

3.1 Основні критерії перевірки випадкових спостережень

1) Крітерій «хи-квадрат».

Припустимо, що кожне спостереження може належати одній з k категорій. Проводимо n незалежних досліджень. Нехай p_s- ймовірність того, що кожне спостереження відноситься до категорії s, і нехай Y_s-число спостережень, які дійсно відносяться до s. Утворюємо статистику: V = $\sum_{s=1}^{k} \frac{(Y_s - np_s)^2}{np_s}$

Зводячи в квадрат $(Y_s - np_s)^2$=Y_s^2-2 np_sY_s+$n^2p_s^2$ і з огляду на той факт. що Y1+Y2+…+Yk=n, p1+p2+…+pk=1, отримуємо формулу V = $\frac{1}{n}\sum_{s=1}^{k}\left(\frac{{Y_s}^2}{p_s}\right) - n$, яка часто спрощує обчислення V.

Повертаючись до питання «Чому дорівнює прийнятне значення V?». Його визначають за допомогою таблиць, які дають значення «χ^2-розподілу с u ступенями свободи» для різних значень u.

2) Критерій Колмогорова-Смирнова.

Розглянемо випадкові величини, які приймають нескінченно багато значень, такі як випадкові дроби (випадкові дійсні числа між 0 і 1). Хоча тільки кінцева безліч дійсних чисел може бути представлена в комп'ютері, ми хочемо, щоб вони вели себе подібно до реальних чисел в інтервалі [0,1).

В теорії ймовірностей і математичній статистиці одні і ті ж позначення використовуються, коли випадкова величина приймає кінцеве або нескінченне число значень. Щоб задати розподіл значень випадкової величини X, слід зробити це в термінах функції розподілу F(x), де

F(x) = Pr(X≤x)=ймовірності, що (X≤x).

Якщо виконати n незалежних спостережень випадкової величини X і отримати значення X1, X2, ..., Xn, то можна буде побудувати емпіричну функцію розподілу Fn(x), де

$$Fn(x)=\frac{\text{число X1},X2,\ldots,Xn,\text{таких,що вони} \leq x}{n}$$

Коли n стає більше, Fn(x) більш точно наближає F(x).

Критерій Колмогорова-Смирнова може використовуватися, коли функція F(x) неперервна (не має стрибків і розривів). Він заснований на різниці між F(x) і Fn(x). Погант джерело випадкових чисел буде давати таку емпіричну функшію, яка погано наближає передбачувану функцію розподілу F(x).

Щоб побудувати критерій Колмогорова-Смирнова, утворюємо такі статистики:

$K_n^+=\sqrt{n}$ max (Fn(x)-F(x)); $K_n^-=\sqrt{n}$ max (F(x)-Fn(x)), де -∞<x<+∞.

Тут K_n^+ визначає найбільше з відхилень, коли Fn більше F, а K_n^- - Fn менше F.

Множник $\sqrt{n}$ з'являється з того, що для фіксованого x стандартне відхилення Fn(x) пропорційно $1/\sqrt{n}$; отже, множник $\sqrt{n}$ збільшує статистики таким чином, що це стандартне відхилення K_n^+ і K_n^- не залежить від n.

Формули наведені вище в тому вигляді, в якому вони записані, не зовсім підходять для обчислень на комп'ютері, так як формально потрібно знаходити максимум по нескінченному безліччю значень x. Але, з огляду на той факт, що F(x) зростає, і то що Fn(x) зростає тільки в кінцевому безлічі точок, можна припустити просту процедуру для підрахунку статистик K_n^+ і K_n^-:

Крок 1. Отримати незалежні спостереження X1, X2, …, Xn.

Крок 2. Впорядкувати спостереження так, що б вони розташовувалися в порядку зростання (побудувати варіаційний ряд): X1<X2<…<Xn.

Крок 3. Потрібні статистики зараз задаються формулами

$K_n^+=\sqrt{n}$ max($\frac{j}{n}$-F(Xj)), 1≤j≤n; $K_n^-=\sqrt{n}$ max (F(Xj)- $\frac{j-1}{n}$), 1≤j≤n

3.2 Емпіричні критерії

Розглянемо декілька критеріїв, які застосовуються для визначення випадковості послідовності.

Кожен критерій застосовується до послідовності

<Un>=U0, U1, U2, … (1)

дійсних чисел, які, як передбачається, незалежні і рівномірно розподілених між 0 і 1. Деякі з критеріїв призначені безпосередньо для

цілочисельних послідовностей, а не послідовностей дійсних чисел. У такому випадку замість неї використовується допоміжна послідовність

<Yn>=Y0,Y1,Y2,…, (2) певна правилом Yn=[dUn]

Це послідовність цілих чисел, які, як передбачається, незалежні і однаково розподілені між 0 і d-1 (інакше кажучи, ймовірність, що випадкова величина прийме значення k,k=0,…,d-1). Перерахуємо найбільш використовувані емпіричні критерії:

1. Критерій рівномірності (критерій частот).

Перша вимога, що пред'являється послідовності (1), полягає в тому, що її члени повинні бути рівномірно розподілені між 0 і 1. У загальному випадку це можна перевірити, використовуючи один з вищенаведених критеріїв: хі-квадрата або Колмогорова-Смирнова з F(x)=x. Також можна використовувати послідовність (2) замість (1), де d-підходяще число. Для кожного r, 0≤r≤d, порахуємо кількість випадків коли Yi=r, для 0≤i≤n, а потім застосуємо хі-квадрат критерій, приймаючи k=d і ймовірності Ps=1/d для кожної категорії.

2. Критерій серій.

Більш загальна вимога до послідовності полягає в тому, щоб пари послідовних чисел були рівномірно розподілені незалежним чином. У критерії серій підраховується число випадків, коли пара (Y2i,Y2i+1)=(q,r) для 0≤i≤n. Така операція існує для кожної пари цілих чисел (q,r), таких, що 0≤q,r<d. Потім застосовуємо хі-квадрат критерій до цих k=d^2 категоріям, де 1/d^2- ймовірність віднесення пари чисел до кожної з категорій. Цей критерій також можна узагальнити і для трійок, четвірок і тд. замість пар. Тоді значення d необхідно істотно зменшити для того, щоб число категорій не вийшло занадто великим.

3. Критерій інтервалів.

Цей критерій іспользустся для перевірки довжини «інтервалів» між появою Ui, на певному відрізку. Якщо α і β-два дійсних числа таких, що

$0 \leq \alpha < \beta < 1$, то розглянемо довжини підпослідовностей Ui,Ui+1,…,Ui+r лежить між α і β, а інші U, не лежать між цими числами.

4. Покер критерій (критерій розбиву).

«Класичний» покер-критерій розглядає n груп по п'ять послідовних цілих чисел {Y5j,Y5j+1,Y5j+2,Y5j+3,Y5j+4}для кожного $0 \leq j < n$ і перевіряє, які з наступних семи комбінацій відповідають таким п'ятіркам чисел (порядок не має значення).

Всі числа різні: abcde

Одна пара: aabcd

Дві пари: aabbc

Три числа одного виду: aaabc

Повний набір: aaabb

Чотири числа одного виду: aaaab

П'ять чисел одного виду: aaaaa

χ^2-критерій заснований на підрахунку числа п'ятірок в кожній категорії.

5. Критерій збирання купонів.

Цей критерій співвідноситься з покер-критерієм, так як критерій інтервалів співвідноситься з критерієм частот. Використовується послідовність Y0, Y1,... і знаходяться довжини відрізків Yj+1,Yj+2,..,YJ+r, які містять повний набір цілих чисел від 0 до d-1.

6. Критерій перестановок.

Розділимо послідовність на виході на n груп по t елементів в кожній, тобто (Ujt,Ujt+1,...,Ujt+t-1) для $0 \leq j < n$.Елементи в кожній групі можна упорядкувати t! різними способами. Підраховується число груп з будь-яким можливим порядком і застосовується χ^2-критерій з k=t! можливими категоріями і ймовірністю 1/t! для кожної категорії.

7. Критерій монотонності.

У послідовності можна перевіряти розподіл висхідних і низхідних серій. Мається на увазі перевірка розподілів довжин монотонних частин заданої послідовності (відрізки зростання або зменшення).

8. Критерій "максимум-t".

Позначимо Vj-max(Ujt,Ujt+1,…,Ujt+t-1) для 0≤j<n. Застосуємо критерій Колмогорова-Смирнова до послідовності V0,V1,...,Vn-1. Таким чином перевіримо гіпотезу про те, що функція розподілу Vj, дорівнює F(x)=x^t, 0≤x<1. Можна також застосувати критерій Колмогорова-Смирнова до послідовності $V_0^t, V_1^t,\ldots, V_{n-1}^t$, перевіряючи гіпотезу про рівномірний розподіл. Для обґрунтування критерію потрібно показати, що функція розподілу Vj дорівнює F(x)=x^t. Імовірність того, що max (U1,U2,…,Ut)≤x, равна ймовірності того, що одночасно U1≤x,U2≤x,…,Ut≤x, яка, в свою чергу, дорівнює добутку ймовірностей Uk≤x при k=,…,t а саме xx…x=x^t.

9. Критерій конфліктів.

Припустимо, є m урн, і помістимо в них навмання n куль, причому m памного больпе n. Більшість куль потрапить в порожні урни, але якщо куля потрапляє в урну, в якій вже міститься хоча б один шар, то будемо говорити, що стався «конфлікт». Критерій конфліктів підраховує число конфліктів, і генератор відповідає цьому критерію, якщо не виникає занадто багато або занадто мало конфліктів. Критерій конфліктів можна використовувати для того, щоб оцінювати генератор випадкових чисел для рядків великих розмірностей. Наприклад, коли m=2^{20} і n=2^{14}, можна перевіряти 20-мірну випадковість генератора випадкових чисел, поклавши d=2 і сформувавши 20-мірний вектор Vj=(V20j,V20j+1,…,V20j+19) для 0≤j<n. Ми зберігаємо таблицю з m=2^{20} двійкових розрядів для визначення конфліктів і один двійковий розряд для кожного можливого значення координати вектора Vj Спочатку в усі 2^{20} двійкових розрядів таблиці занесемо 0; потім для кожного Vj, якщо в відповідний двійковий розряд вже занесена 1, реєструємо конфлікт,

в іншому випадку заносимо в двійковий розряд 1.

Що б вирішити, чи проходить послідовність випробування, використовуються таблиці процентних точок.

10. Критерій проміжків між днями народження.

Помістимо n куль в m урн, де під урнами будемо мати на увазі дні року, а під кулями - дні народження. Припустимо, що (Y1,...,Yn) - це дні народження, де $0 \le Yk < m$. Розташуємо їх у порядку неспадання $Y(1) \le \ldots \le Y(n)$, визначимо n «проміжків» $St=Y(2)-Y(1),\ldots,Sn-1=Y(n)-Y(n-1),Sn=Y(1)+m-Y(n)$ і розташуємо проміжки в такому порядку:$S(1) \le \ldots \le S(n)$. Нехай R - число рівних проміжків, а саме-число індексів j, таких, що $0 \le j < n$ і $S(j)=S(j-1)$.

3.3 Теоретичні критерії

Незважаючи на те, що кожен генератор випадкових чисел можна перевірити за допомогою емпіричних критеріїв, краще мати апріорний критерій, а саме - теоретичний результат, за допомогою якого можна заздалегідь сказати, наскільки гарні будуть перевірки генератора. Такі теоретичні результати допомагають зрозуміти методи генерування набагато краще, ніж емпіричні методи проб і помилок. Якщо можна передбачити результати певних перевірок заздалегідь, перш ніж генерувати випадкові числа, то є більше можливостей належним чином вибрати a, m і b.

Отримано поки далекі від загальних результати для статистичних критеріїв, які можна застосовувати до повних періодів. Проте, статистичні критерії набувають сенсу, коли вони застосовуються до всього періоду. Дослідження всього періоду визначатиме глобальну «невипадковість» послідовності, тобто неправильна поведінка дуже великих вибірок.

3.4 Спектральний критерій

Спектральний критерій-особливо важливий метод перевірки якості конгруутних генераторів випадкових чисел. Це найбільш потужний, відомий на сьогоднішній день, критерій перевірки випадковості. Спектральний критерій має властивості як емпіричних, так й теоретичних критеріїв. Він схожий на емпіричні, тому що для отримання результату вимагає обчислень на комп'ютері, на теоретичні, оскільки перевіряє властивості послідовності на повному періоді. Найбільш важливі випробування для перевірки, наскільки випадковою буде послідовність, пов'язана з властивостями спільного розподілу t послідовних елементів послідовності, і спектральний критерій як раз і використовується для перевірки гіпотез про ці розподіли. Якщо задана послідовність <Un>=U0,U1,U2,…, з періодом m, то для побудови критерію необхідно проаналізувати безліч всіх m точок

{(Un,Un+1,…,Un+t-1)|0≤n≤m} (3) в t-мірному просторі.

Нехай 1/V2-максимальна відстань між лініями всіх сімейств паралельних ліній, які проходять через двовимірні точки {(x/m,s(x)/m)}. Назвемо V2 двовимірной точністю генератора випадкових чисел, так як пари послідовних чисел мають гарну структуру, яка особливо хороша щодо V2. Аналогічно, нехай 1/V3-максимальна відстань між площинами з сімейства паралельних площин, що проходять через всі крапки {(x/m,s(x)/m),s(s(x))/m)}; назвемо V3 точністю в трьох вимірах. t-мірна точність Vt равна величині, зворотній мінімальній відстані між гіперплоскостямі з сімейств паралельних (t-1)-мірних гіперплоскостей. Коли необхідно перевірити чи будуть t послідовних значень незалежні, комп'ютерні генератори будуть урізати їх до InVt двійкових розрядів, де Vt убуває зі зростанням t. На практиці така зміна точності цілком прийнятно. Однак, коли в додатках потрібен генератор випадкових чисел, який забезпечує отримання послідовності, дуже близькою до випадкової, прості конгруентні генератори для цього не дуже підходять. Замість них потрібно використовувати генератор з дліним періодом, навіть

якщо насправді генерувати тільки малу частину періоду. Якщо довжину періоду звести в квадрат, то буде зведена в квадрат і точність у великих вимірах, тобто ефективне число точних розрядів подвоїться.

У ряді робіт, опублікованих в 70-і роки, Гаральд Нідеррейтер показав, як проводити дослідження t-мірних векторів (3) за допомогою експоненційних сум. Одним з основних результатів його теорії було наступне: він показав, що генератор випадкових чисел проходить перевірку за допомогою критерію серій для декількох вимірювань, якщо цей генератор витримує перевірку спектральним тестом, навіть коли замість повного періоду розглядається велика його частина.

Введемо поняття-розкид в t вимірах, тобто величина, яку визначимо як максимум між середнім числом і реальним числом t-мірних векторів (xn,xn+1,…,xn+t-1), що потрапляють в гіперпрямоугольну область з усіх вищевказаних областей. Точніше, якщо <xn>-послідовність цілих чисел в області 0≤xn<m, визначимо об'емR D - max (+) де R-області виду тут a, і в - цілі числа з областей 0sa, s B sm для 1sist, 0б'ем R. очевидно, дорівнює (в - a,) ... (У, -a,). Щоб знайти розкид D, розглянемо всі безлічі R і знайдемо таке, у якого найбільше або найменше число точок виду (x,, x, x).

Верхня межа розкиду може бути знайдена за допомогою експоненційних сум. Нехай w -е. первісний т-й корінь з одиниці. Якщо (x, ..., x,) і (y, - два вектора з компонентами з області 0s x ,, y, < т, то справедливо рівність [m ', якщо (x, ... x,) - (yy,).

Тому число векторів (x,, x,, .x) в R, коли область R визначена в (5), може бути виражена таким чином

Коли в цій сумі i, - i, -0, то вона дорівнює N т, помноженому на обсяг R. Следоватсльно, D можна виразити, як максимум по До вираження.

Виконавши перетворення можна отримати, що де - max Легко довести, що сума g (i, .. i,) буде маленькою, якщо тільки не виконується g (i, ..., i,) 0 (по модулю m).

Отже, розкид D буде малим, коли пройдсна перевірка спектральним тестом.

4. ПОБУДОВА ОДНОСПРЯМОВАНОЇ ФУНКЦІЇ

У ряді робіт С.П. Варбанець вивчив структуру послідовності псевдовипадкових чисел, згенерованої конгруетним інверсним генератором:

y_{n+1}=ay_n+b(mod p^m)

Його основна теорема має вигляд:

Нехай заданий інверсний генератор під умовою (y0,p)=(a,p)=1, b=p^{υ}b0, $^{\upsilon}\geq$1, (b0,p)=1. Тоді існують многочлени f0(bc) і f1(bc), такі, що

$y_{2k}\equiv a_0+a_1$k+$a_2p^{2^{\upsilon}}k^2+\ldots$, $a_1\equiv$0 (mod p), $a_j\equiv 0(p^{3^{\upsilon}})$, j=3,4,…

$y_{2k+1}\equiv b_0+b_1$k+$b_2p^{2^{\upsilon}}k^2+\ldots$, $b_1\equiv$0 (mod p), $b_j\equiv 0(p^{3^{\upsilon}})$, j=3,4,…

Причому послідовність $\{\frac{y_n}{p^m}\}$, n=0,1,…, проходить серіальний тест на равнорозподіленність і передбачуваність.

Слідство: послідовна $\{\frac{y_n}{p^m}\}$, породжена рекурсією, може застосовуватися в криптографії.

Ми покажемо, як результат цієї теореми можна застосувати для побудови односпрямованої функції з $Z_{p^{m-\upsilon}}$ в $Z^*_{p^m}$, де $Z_{p^{m-\upsilon}}$-числа 0,1,…, $p^{m-\upsilon}$-1, $Z^*_{p^m}$ складається з чисел 1,2,…, p^m, які взаємно прості з p.

Нехай n Є $Z_{p^{m-\upsilon}}$. Тоді

f(n) = $\begin{cases} y_{2k}, \text{якщо } n=2k, \\ y_{2k+1}, \text{якщо } n=2k+1, \end{cases}$ k=0,1,…

Складність відновлення числа n за значенням f(n) є наслідок равнорозподіленністі і статистичної незалежності (непередбачуваності) послідовності значень y_n.

5. ПРОГРАМНА РЕАЛІЗАЦІЯ

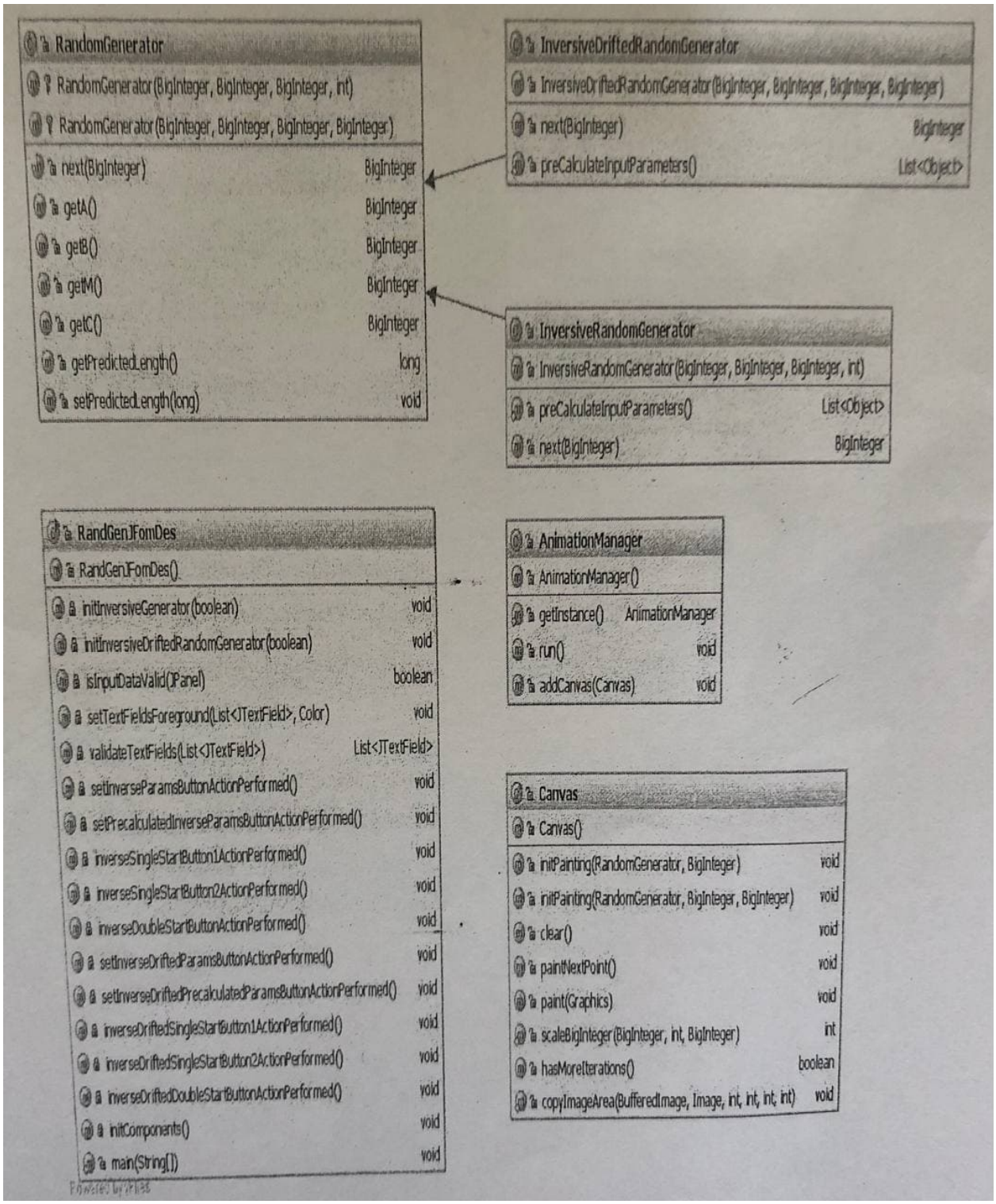
4.1 Загальні відомості

Програма написана на мові програмування Java версії 1.6. Для побудови графічного інтерфейсу використовувалася бібліотека Swing. Вибір мови обумовлений декількома критеріями. Мова Java має потужну бібліотеку для математичних розрахунків, зокрема вбудовану підтримку роботи з числами необмеженого діапазону, алгоритми пошуку простих чисел і інверсії по модулю. Крім цього Java надає кошти для побудови сучасного графічного інтерфейсу користувача, а так само має встроену підтримку багатопоточності, що дозволяє реалізувати відображення результатів роботи в простій і наочній формі, з можливістю візуалізації процесу генерації псевдовипадкових послідовностей в динаміці. На додаток до цього для мови Java існують просунуті середовища розробки (IDE) і побудови графічного інтерфейсу, такі як NetBeans та Eclipse, на порядок перевершують аналогічні IDE для інших мов.

Дотримуючись парадигмі об'єктно-орієнтованого програмування, програма складається з набору класів, розміщених в наступних пакетах:

com.fit.random.generators	Містить класи моделі програмного уявлення конгруенціальних генераторів.
com.fit.random.visual	Містить класи уявлення.

4.2 Діаграма класів



4.3 Опис алгоритму

Точкою входа в програму є метод main() метод класу RandomGeneratorFrame. Цей клас містить код для створення всього графічного інтерфейса.

```
public static void main (String [] args) {
     try {
          UIManager.setLookAndFee1 (new WindowsLookAndFeel
     ());
     } catch (UnsupportedLookAndFee1Exception e) {
          e.printStackTrace ();
     }
     new RandGenJFomDes().setvisib1e (true);
}
```

У методі main() відбувається настройка параметрів відображення графічних компонент і створюється об'єкт класу RandomGeneratorFrame. У конструкторі відбувається ініціалізація полів класу, зокрема:

```
private InversiveRandomGenerator inversiveRandomGenerator,
private InversiveDrifedRandomGenerator
inversiveDrifedRandomGenerator;
```

Це посилання на об'єкти класів, що представляють собою моделі конгруенціальних генераторів псевдовипадкових чисел. В основі ієрархії класів моделі лежить абстрактний клас RandomGenerator. Цей клас містить поля і методи, загальні для всіх конгруснціальних генераторів, наприклад посилання на об'єкти Biglnteger, що зберігають значення поточних a, b, m, c, початкового і поточного значення аргументу для кожного конкретного генератора, а так само геттери і сеттери для цих полів. Також цей клас містить абстрактний метод: `public abstract Biglnteger next (BigInteger current);` призначений для отримання наступного елемента послідовності. Класи InversiveDrifedRandomGenerator і InversiveRandomGenerator успадковуються від RandomGenerator і є моделлю інверсного і інверсного зі зміщенням відповідно. Метод next() класу InversiveRandomGenerator:

```
public Bigtnteger next (Biglnteger current){
     BigInteger m - getM ();
     1f (current.gcd(m).1ongValue()==1) return
current.modInverse(m).multiply (getA ()).add(getB()).mod
(m);
else return getB().mod(m);
}
```

Як видно, це формула (7), записана на мові програмування. Цей клас успадковує всі поля та методи класу RandomGenerator. Аналогічний метод next() класу InversiveDriftedRandomGenerator:

```
public BiglInteger next (BigInteger current){
     B1gInteger bigCounter B1glnteger=new BigInteger (new
Integer(++iterationcounter.toString());
     if(current.gcd(getM()).longValue()==1) return
current.modInverse
(getM()).multiply(getA()).add(getB()).add(getC().multiply(bigCou
nter)).mod(getM());
     else return
getB().add(getC().multiply(bigCounter)).mod(getM());}
```

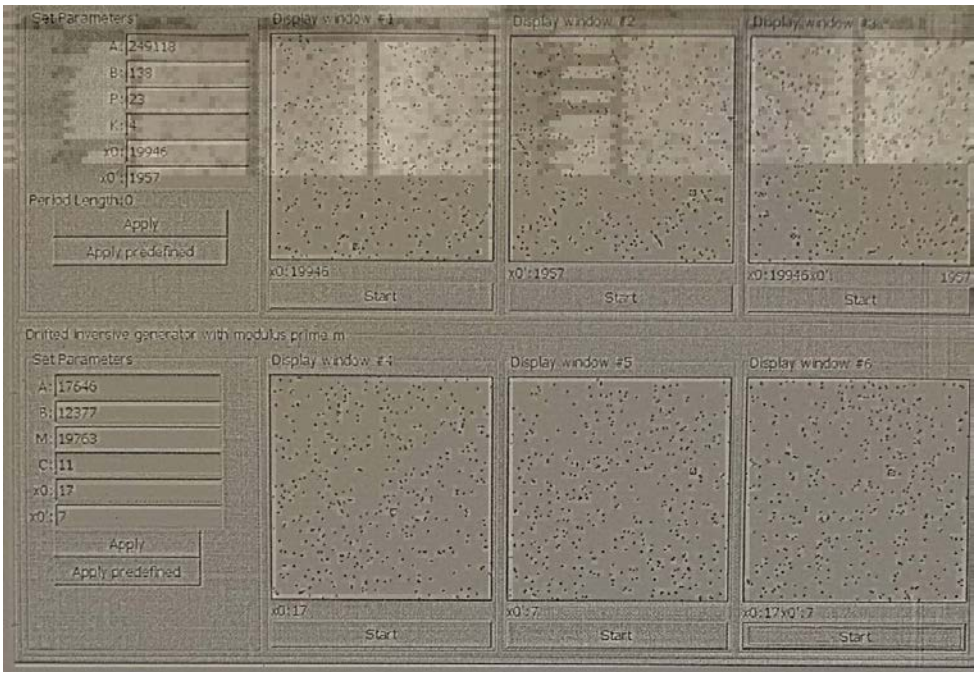
представляє собою формулу x_{n+1}=(a*x_n^{-1}+b+c*n) mod p якщо x_n≠0; x_{n+1}=b+c*n якщо x_n=0.

В класі InversiveRandomGenerator є метод, який конструює інверсний генератор з максимальним періодом по теоремі Чоу:

```
public static List<Object> preCalculateInputParameters() {
     Random random=new Random ();
     BigInteger p;
     int k;
     int j;
     long predictedLength;
     do {
          do {
               k=random.nextInt(5)+3;
```

```
                j=1;
            } while (k<=j);
            P=BigInteger.valueOf(random.nextInt(100)+10).nextP
robablePrime();
            predictedLength=p.pow(k-j).longValue()*2;
        } while (predictedLength<1000||predictedLength>1000);
        int d=random.nextInt(10)+1;
        BigInteger b=p.pow(j).multiply(BigInteger.valueOf(d));
        long a;
        long x0;
        BigInteger m=p.pow(k);
        do {
            a=random.nextLong()%value+value;
            x0=random.nextLong()%value+value;
        } while (p.gcd(bigInteger.valueOf(a*x0*(a-
x0*x0))).longValue()!=1);
        List<Object> result=new ArrayList<Object>();
        result.add(a);
        result.add(b);
        result.add(p);
        result.add(k);
        result.add(x0);
        result.add(x0);
        return result;
}
```

Значення для параметрів інверсного генератора вибираються на основі теореми 5 (С). Це дозволяє домогтися порівняно великої довжини періоду по відношенню до модуля при відносно невеликій кількості обчислень. Таким чином, основною функцією абстрактного класу RandomGenerator є завдання кореня ієрархії і загального інтерфейсу класів-генераторів в даній моделі. Крім цього, клас реалізує класичний шаблон проектування «Фабрика» -його фабричні методи createLinearGenerator() і createInversiveGenerator() повертають об'єкт одного із спадкоємців цього класу.



(рис. 1)

На малюнку 1 представлений вид головного вікна програми. У лівій частині вікна розташовані панелі інформації про генератори. Вони відображають основні характеристики поточних генераторів. При натисканні «Apply» буде сконструйований відповідний генератор. Праворуч від панелей генераторів розташовуються 6 спеціальних графічних компонент- полотен, які служать для відображення генеруємих точок. При натисканні кнопки «Start» запускається процес генерування і відображення точок на відповідному полотні, починаючи з зазначеного в компоненті значення хо. У програмі полотна реалізуються при допомозі класу Canvas. Об'єкти цього класу інкапсулюють в собі код відтворення. При виклику методу

```
Public void paintNextPoint () {
    …
    graphics.drawLine (xo,y0,x0+1,y0);
    graphics.drawLine (xo,y0+1,x0+1,y0+1);
    copyImageArea (image,copy,x0-2,yo-2,6,6);
    graphics.setColor (Color.BLUE);
    graphics.drawRect (x0-2,y0-2,5,5);
    repaint (x0-2,y0-2,6,6);}
```

відбувається розрахунок двох наступних значень елементів послідовності, що утворюють точку, перетворення координат, отрисовка отриманої точки і рамки навколо неї в буфер і виклик методу поновлення ділянки графічного компонента. Процес генерування і відтворення точок відбувається в динаміці з інтервалом між точками в 1 секунду, що дає можливість спостерігачеві оцінити властивості генерируємої послідовності. Періодичність процесу досягається за рахунок того, що метод paintNextPoint() у відповідного об'єкта Canvas викликається з окремого потоку. Керуючий отрисовкой код знаходиться у класі AnimationManager. Цей клас реалізує шаблон проектування «Одинак».

```
    private static final AnimationManager instance=new
AnimationManager();
    public static AnimationManager getInstance() {return
instance;}
```

В конструкторі класа створюється новий потік:

```
public AnimationManager() {
     isRunning=true;
     canvases=new HashMap<Canvas, Canvas>();
     (new Thread(this)).start();}
```

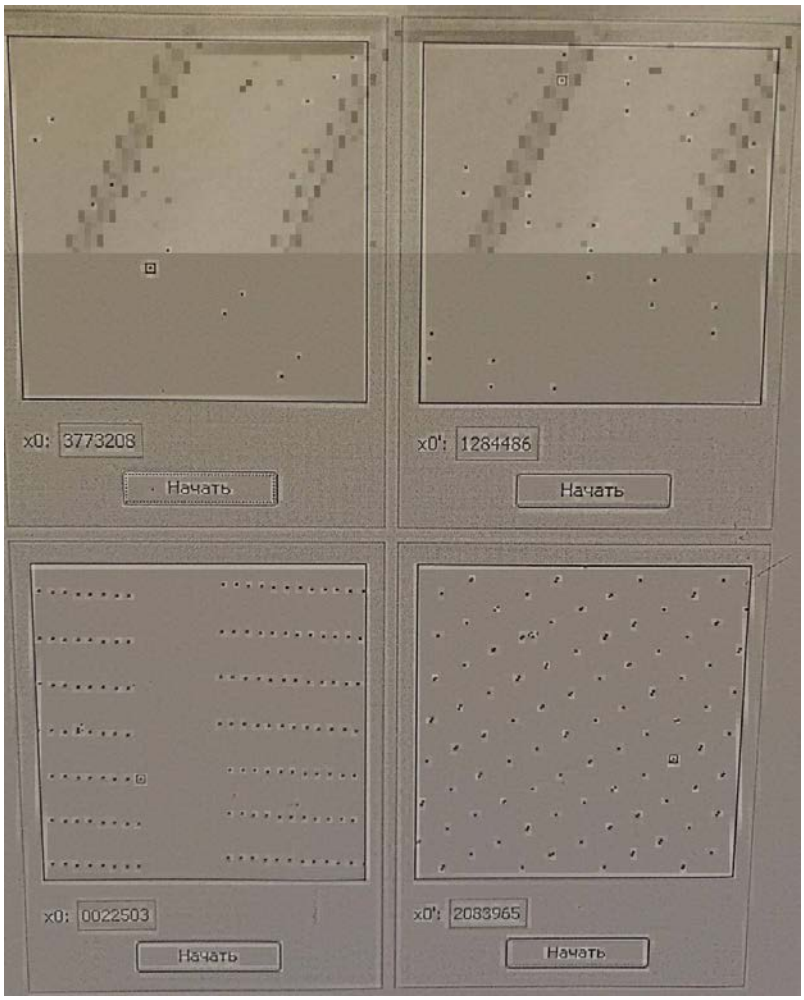
який виробляє періодичність отрисовки точок на всіх «активованих» полотнах. Список полотен зберігається в хеш-таблиці canvases, яка є полем класу. Нові полотна додаються туди за допомогою методу:

```
public void addCanvas(Canvas canvas) {
     synchronized(canvases) {
          canvases.put(canvas,canvas);}}
```

Таким чином в програмі існують два потоки виконання: системний потік обробки подій (EventDispatchThread) і потік анімації.

Взаємодія потоків здійснюється через поле canvases класу AnimationManager, тому робота з цим полем синхронізована.

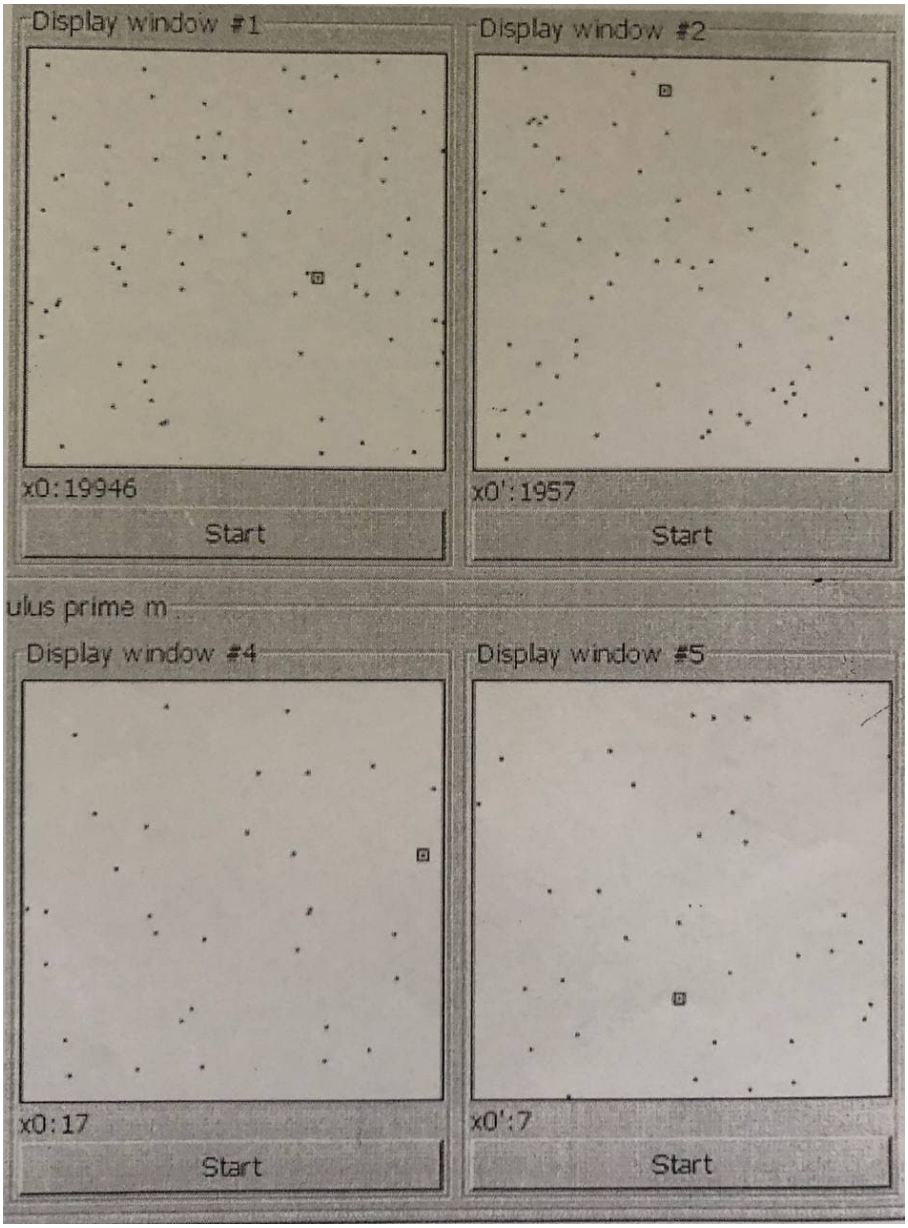
Спостерігаючи і аналізуючи динаміку процесу генерування точок, можна судити про якість тестованого генератора. Для передбачуваних генераторів в процесі виникнення нових точок можна вловити деякі закономірності і передбачити, де саме з'явиться наступна точка.



(рис. 2)

На малюнку 2 показаний зовнішній вигляд полотен для передбачуваних генераторів. Як видно з малюнка, у взаємному розташуванні точок існують чіткі закономірності. Така картина спостерігається для більшості лінійних генераторів і для деяких інверсних. У процесі дослідження було виявлено, що при малих значеннях параметрів a і/чи x0 інверсний генератор стає передбачуваним.

Цей факт пояснюється тим, що в цьому випадку кожен k-ий член буде слабо відрізнятися від до k-2-го і всі крапки будуть мати дуже близькі стосунки координат і, фактично, будуть лежати на одній прямій.



(рис. 3)

Малюнок 3 ілюструє загальний вигляд полотен для непередбачуваних генераторів.

Як видно з малюнків, умова рівномірності виконується для всіх генераторів, не залежно від їх передбачуваності.

ВИСНОВОК

В ході виконання роботи були вивчені сучасні методи генерації псевдовипадкових послідовностей за допомогою конгруентних генераторів. Шляхом численних експериментів було встановлено залежність якостей генерованої послідовності від параметрів генератора. В результаті проведеної роботи були зроблені наступні висновки.

Інверсні генератори мають меншу довжину періоду по відношенню до величини модуля і вимагають більшої кількості обчислень для отримання чергового елементу. Отримувані послідовності мають рівномірний розподіл і достатню випадковість (при великих значеннях параметрів a і x0 по відношенню до модуля). Це робить інверсні генератори застосовними для використання в криптографічних додатках. Залежність результатів від змінних a, b, c є не настільки видимою як у випадку з лінійним генератором, однак їх вибір є основним завданням. Змінна a має найбільший внесок в «випадковість» послідовності і її період. При малих значеннях період послідовності буде невеликий. Досягнення максимальних періодів при різних значеннях змінних a, b, c було вивчено Стівеном Чоу і перевірено в дипломній роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. D. H. Lehmer, Proc. 2nd Symp. on Large-Scale Digital Calculating Machinary, Cambridge, Mas.: Harvard University Press (1951), 141-146.

2. W. E. Thompson. Comp. J. 1p. (1980), 83-86.

3. A. Rotenberg, JACM 7 (1960), 75-77.

4. M. Greenberger, JACM 8 (1961), 383-389.

5. Hull, Dobell, SIAM Review 4 (1962), 230-254.

6. W-S. Chou, On period lengths of inversive congruential recursions, ACTA ARITHMETICA LXXIII.4 (1995), 325-341.

7. J. Eichenauer and J. Lehn, A non-linear congruential pseudorandom number generator, Statist. Hefte 27 (1986), 315-326.

8. H. Niederreiter, Finite fields and their applications; Contributions to General Algebra, Vol. 7, Vienna, Teubner, Stuttgart (1990), 200-205.

9. W.-S. Chou, On inversive maximal period polynomials over finite fields, Appl. Algebra Engrg. Comm. Comput. 6 (1995), 245-250.

10. M. Flahive and H. Niederreiter, On inversive congruential generators for pseudorandom numbers; Finite Fields, Coding Theory, and Advances in Communications and Computing, G. L. Mullen and P. J.-S. Shiue (eds.), Marcel Dekker, New York (1992), 75-80.

11. J. Eichenauer, J. Lehn and A. Topuzoglu, A nonlinear congruential pseudorandom number generator with power of two modulus, Math. Comp. 51 (1988), 757-759.

12. W.-S. Chou, The period lengths of inversive pseudorandom vector generators, Finite Fields Appl. 1 (1995), 126–132.

13. K. Huber, On the period length of generalized inversive pseudorandom generators, Appl. Algebra Engrg Comm. Comput. S (1994), 255–260.

14. H. Niederreiter and I. Sparlinski, Exponential sums and the distribution of inversive congruential pseudorandom numbers with prime-power modulus, Acta Arithm. 90 (2000), 89-98.

15. S. Varbanets, On inversive congruential generator for pseudorandom numbers with prime-power modulus, Ann. Univ Sci. Budapest. Sect. Computatorica, XXVIII (2010), 5-7.

16. D Knuth, The Art of Computer Programming, vol. 2 Seminumerical Algorithms, Boston, Mass Addison-Wesley (2006), 832-879.