Дослідження алгоритмів генерації гама шифру

# Ціль: зробити порівняльний аналіз алгоритмів генерації гама шифру

Теоретичний матеріал

Під *гамуванням* розуміють процес накладення за визначеним законом гами шифру на відкриті дані. *Гама шифру -* це псевдовипадкова послідовність, вироблена за заданим алгоритмом для шифрування відкритих даних і розшифрування зашифрованих даних.

Процес шифрування полягає в генерації гами шифру і накладенні отриманої гами на вихідний відкритий текст оборотним образом, наприклад з використанням операції додавання по модулю 2.

При шифруванні методом гамування як ключ використовується випадковий рядок бітів, що поєднується з відкритим текстом, також представленим у двоїчному виді за допомогою побітового додавання по модулю 2, і в результаті виходить шифрований текст. Генерування непередбачених двоїчних послідовностей великої довжини є однією з важливих проблем класичної криптографії. Для вирішення цієї проблеми широко використовуються генератори двоїчних псевдовипадкових послідовностей.

Псевдовипадкові ряди чисел, які генеруються, часто називають гамою шифру чи просто гамою. Звичайно для генерації послідовності псевдовипадкових чисел застосовують комп'ютерні програми, що, хоча і називаються генераторами випадкових чисел, насправді видають детерміновані числові послідовності, що по своїх властивостях дуже схожі на випадкові .

До криптографічне стійкого генератора псевдовипадкової послідовності чисел (гами шифру) пред'являються три основних вимоги:

• період гами повинний бути досить великим для шифрування повідомлень різної довжини;

• гама повинна бути практично непередбаченою, що означає неможливість пророчити наступний біт гами, навіть якщо відомі тип генератора і попередній шматок гами;

• генерування гами не повинне викликати великих технічних складностей.

Довжина періоду гами є самою важливою характеристикою генератора псевдовипадкових чисел. По закінченні періоду числа почнуть повторюватися, і їх можна буде пророчити. Необхідна довжина періоду гами визначається ступенем закритості даних. Ніж довше ключ, тим сутужніше його підібрати. Довжина періоду гами залежить від обраного алгоритму одержання псевдовипадкових чисел.

Щоб гама вважалася непередбаченою, тобто істинно випадковою, необхідно, щоб її період був дуже великим, а різні комбінації бітів визначеної довжини були рівномірно розподілені по всій її довжині.

Третя вимога обумовлює можливість практичної реалізації генератора програмним чи апаратним шляхом із забезпеченням необхідної швидкодії.

З відомих процедур генерації послідовності псевдовипадкових цілих чисел найбільше часто застосовуються конгруентні датчики і датчики М-послідовностей.

В даний час найбільш доступними й ефективними є конгруентні генератори ПСП. Для цього класу генераторів можна зробити висновок про те, якими властивостями володіють вихідні сигнали цих генераторів з погляду періодичності та випадковості.

Одним з гарних конгруентних генераторів є лінійний конгруентний датчик ПСЧ. Він виробляє послідовності псевдовипадкових чисел *T(i),* які описуються співвідношенням

*Т*(*i+*1) = ( *A\*T*(*i*) +*C* ) *mod m,*

де *А* и *С* — константи;

*Т*(0) — вихідна величина, обрана як породжуюче число. Очевидно, що ці три величини й утворять ключ.

Такий датчик генерує псевдовипадкові числа з визначеним періодом повторення, що залежить від обраних значень *А* и *С*. Значення *m* звичайно встановлюється рівним *2п ,* де *п —* довжина машинного слова в бітах. Датчик має максимальний період *М* до того, як послідовність, яка генерирується почне повторюватися. Через причину, відзначену раніше, необхідно вибирати числа *А* и *С* такі, щоб період *М* був максимальним.

Лінійний конгруентний датчик ПСЧ має максимальну довжину *М* тоді і тільки тоді, коли *З* — непарне, і *A* *mod* 4=1. Для шифрування даних за допомогою датчика ПСЧ може бути обраний ключ будь-якого розміру. Наприклад, нехай ключ складається з набору чисел *x*(*j*) розмірністю *b,* де *j =* 1, 2, ..., *п.* Тоді створювану гаму шифру *G* можна представити як об'єднання непересічних множеств *H*(*j*)

М-послідовності являють собою лінійні рекурентні послідовності максимального періоду, формовані *k-розрядними* генераторами на основі регістрів зсуву. На кожнім такті біт, що надійшов, *зсуває* k попередніх і до нього додається їхня сума по модулю 2. Біт, що витісняється, додається до гами.

Строго це можна представити у виді наступних відносин:

*r0*:=*r1*; *r1*:=*r2*; … *rk-1*:=*rk-2*;

*r0*:=*a0r1* ⊕*a1r2* ⊕*…*⊕ *ak-2rk-1*

Г*i*:=*rk*.

Тут *r0*, *r1*,…*rk-1* —*k* однобітних регістрів;

*a0*,*a1*,*…*,*ak-2* — коефіцієнти двоїчного поліному, що неприводиться, *ступеня* k-1;

Г- *i-e* значення вихідної гами.

Для контролю якості сгенерованої гами використовуються різні тести.У якості базових рекомендується використовувати п'ять тестів

1. частотний (монобітний) тест;
2. тест двох бітових серій;
3. тест Поккера;
4. тест серій (загальний);
5. автокореляційний тест.

#### Монобітний тест

Метою монобітного тесту є перевірка того, чи є число символів “0” і “1” у  послідовності приблизно таким, як у випадкової послідовності. Якщо  число символів “0” у послідовності, а  - символів “1”, то параметр ПСЧ



підкоряється  розподілу з одним ступенем свободи (якщо ).

#### Тест двох бітових серій

Метою тесту двох бітових серій є перевірка того, чи збігається число появ серій “00” - , “01” - , “10” - , “11”  - такий, як і у випадкової послідовності. Параметр ПСЧ



підкоряється розподілу з двома ступенями свободи (якщо ).

#### Тест Поккера

Нехай m - ціле число, таке що

 .

Розділимо послідовність Y на k неперекриваючих частин, довжиною m. Нехай i буде число появи послідовності i довжиною m i - того біту. Тест Поккера дозволяє визначити, чи дійсно послідовності довжиною m кожна з'являються стільки ж раз, скільки їх очікується у випадковій послідовності Параметр:



підкоряється  розподілу з  ступенями свободи.

#### Тест серій

Тест серій дозволяє визначити, чи дійсно число нулів чи одиниць (серії) різної довжини в послідовності Y таке ж, як і у випадкової послідовності Бажане число інтервалів довжиною *i* у випадковій послідовності n дорівнює

 .

Нехай k дорівнює найбільшому цілому числу *i*, для якого . Нехай також B і G буде числом блоків і інтервалів відповідно довжиною *i* у Y для кожного *i*, . Тоді параметр



підкоряється розподілу з  ступенями свободи.

#### Автокореляційний тест

Метою автокореляційного тесту є перевірка ступеня зв'язку між  і її сзувами. Нехай d фіксоване ціле число, . Число біт у Y послідовності дорівнює

.

Статистика параметра 

підкоряється  нормальному розподілу, якщо .

### Методичні вказівки

При проведенні лабораторної роботи варто взяти розглянуті алгоритми побудови гами і дослідити вплив породжуючих коефіцієнтів і послідовностей на випадковість появи символу в послідовності. Для цього необхідно підрахувати кількість одиничних символів і всіляких сполучень цих символів. Наприклад: 00, 11, 01, 10, 111, 000 і т.д. По їхньому співвідношенню можна зробити висновок про випадковість появи того чи іншого символу в послідовності.

#### При розгляді впливу обираних коефіцієнтів варто звернути увагу на розмір послідовності, яка генерується до її повторення.

## Хід роботи

1. Усвідомлюється поставлена задача.
2. Розробляється алгоритм програми, що реалізує розглянуті алгоритми генерації гами шифру.
3. Пишеться текст програми.
4. Тестується програмний продукт.
5. Визначається залежність розміру послідовності, яка генерується, від різних значень породжуючих коефіцієнтів.
6. Для перевірки відповідності вимоги до ПСЧ послідовність, яка генерується, тестується двома будь-якими тестами

##### Зміст звіту

1. Алгоритм програми.
2. Таблиця і графіки залежності розміру послідовності, яка генерується, від різних значень коефіцієнтів, що породжують.
3. Таблиця з результатами роботи тестів
4. Висновки.