**Харківський національний економічний університет**

**імені Семена Кузнеця**

**ЗВІТ**

**З ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 5**

**за дисципліною: “ Пототокові алгоритми”**

**Виконав: студент факультету Інформаційних технологій**

**2 курсу, спец. Кібербезпека,**

**групи 6.04.125.010.21.2**

**Бойко Вадим Віталійович**

**Перевірив:**

**Чугай Андрій Михайлович**

**ХНЕУ ім. С. Кузнеця**

**2023**

1. Чим потоковий шифр відрізняється від блокового шифру?

Потоковий шифр та блоковий шифр є двома основними категоріями симетричних шифрів, які використовуються для захисту конфіденційної інформації. Основна різниця між цими двома типами шифрів полягає в тому, як вони обробляють дані для зашифрування.

Потоковий шифр шифрує дані по одному біту або байту за раз, використовуючи потік ключів, тоді як блоковий шифр шифрує дані в блоках фіксованого розміру, оброблюючи їх незалежно один від одного.

2. Як організується шифрування потоку даних змінної довжини?

Шифрування потоку даних змінної довжини організовується за допомогою потокових шифрів. Для шифрування такого потоку даних потрібен ключ та генератор псевдовипадкової послідовності. Генератор псевдовипадкової послідовності забезпечує потік ключів, який використовується для шифрування даних.

Кожен біт чи байт відкритого тексту по черзі шифрується за допомогою потокового шифру, який використовує біти потоку ключів. Шифрований біт чи байт отримується за допомогою операції XOR між бітом чи байтом відкритого тексту та відповідним бітом чи байтом потоку ключів. Цей процес повторюється, поки не будуть зашифровані всі біти чи байти відкритого тексту.

Для забезпечення безпеки шифрування потоку даних змінної довжини важливо, щоб генератор псевдовипадкової послідовності генерував потік ключів, який є досить складним для передбачення. Це забезпечить надійність захисту від різноманітних атак, включаючи атаки з використанням знання деякої частини відкритого тексту та відповідного зашифрованого тексту.

3. Які числа називають псевдовипадковими?

Псевдовипадкове число — елемент отриманої за певним алгоритмом числової послідовності, властивості якої наближаються до випадкової. Негативною особливістю псевдовипадкових чисел (як імітаційної моделі випадкових чисел) є періодичність отриманої послідовності.

4. Яким чином можуть використовуватися для отримання псевдовипадкових чисел регістри зсуву із зворотним зв'язком? Поясніть їхній принцип роботи.

Регістри зсуву з зворотнім зв'язком (LFSR) є одним з способів генерації псевдовипадкових чисел. Принцип роботи полягає в тому, що регістр зсуву містить набір бітів, які зсуваються вправо з певною швидкістю. При цьому, на вихід регістра виводиться значення одного біту, яке потім зворотнім зв'язком вводиться на початок регістра, замінюючи старший біт.

Для генерації послідовності псевдовипадкових чисел, у регістрі зсуву з зворотнім зв'язком обирається початковий стан, який визначається початковим значенням бітів у регістрі. Потім, при зсуванні бітів вправо, з виходу регістра виводиться псевдовипадкове число - один біт послідовності. За допомогою зворотнього зв'язку, це значення потім вводиться на початок регістра, щоб впливати на подальше зсування і визначати наступні значення псевдовипадкової послідовності.

Для того, щоб забезпечити достатню складність і випадковість генерованих чисел, важливо обрати відповідний регістр зсуву з зворотнім зв'язком, який має високий період - тобто, кількість генерованих значень перед повторенням послідовності. Для цього, зазвичай використовуються регістри з простими характеристичними поліномами, які забезпечують максимально можливий період і випадковість генерованих чисел.

5. У чому різниця між генераторами випадкових чи псевдовипадкових чисел?

Генератор випадкових чисел (ГВЧ) і генератор псевдовипадкових чисел (ГПВЧ) - це два різних поняття.

Генератор випадкових чисел - це пристрій або програма, що генерує послідовність чисел, які вважаються випадковими зі статистичної точки зору. Випадковість в цьому випадку визначається на основі природних випадкових процесів, таких як шуми електронних коливань, радіохвиль, геофізичних подій і т. д.

З іншого боку, генератор псевдовипадкових чисел (ГПВЧ) - це алгоритм або програма, яка генерує послідовність чисел, які виглядають випадковими, але насправді вони є детермінованими, тобто залежать від початкових параметрів.

Отже, різниця між ГВЧ і ГПВЧ полягає у тому, що ГВЧ використовує непередбачувані процеси природи, тоді як ГПВЧ використовує детерміновані алгоритми для генерації послідовності чисел, які схожі на випадкові.

6. Чи можна використовувати генератор цих випадкових чисел для отримання гами при потоковому шифруванні?

Так, генератор псевдовипадкових чисел (ГПВЧ) можна використовувати для отримання гамми при потоковому шифруванні. В потоковому шифруванні гамма - це послідовність випадкових бітів, яка генерується генератором псевдовипадкових чисел і використовується для шифрування даних шляхом їх XOR з гаммою.

Генератор псевдовипадкових чисел можна налаштувати таким чином, щоб він генерував випадкові біти, які можуть використовуватися як гамма для потокового шифрування. Проте важливо враховувати, що генератор

псевдовипадкових чисел може мати недоліки, які можуть зробити його менш безпечним для застосування в криптографічних системах. Таким чином, важливо використовувати відповідні алгоритми та ключі для забезпечення безпеки в потоковому шифруванні з використанням генератора псевдовипадкових чисел.

7. З якою криптографічною метою можна використовувати генератори справжніх випадкових чисел?

Генератори справжніх випадкових чисел (ГСВЧ) можна використовувати для багатьох криптографічних цілей, особливо там, де безпека має найважливіше значення. Основна мета використання ГСВЧ полягає в тому, щоб забезпечити криптографічний протокол випадковістю, що є необхідним для захисту від різноманітних атак, таких як атаки перебору або аналізу.

Деякі конкретні застосування генераторів справжніх випадкових чисел у криптографії включають генерацію ключів, створення ініціалізаційних векторів для режимів блочного шифрування, генерацію параметрів елліптичної криптографії, генерацію підписів та ідентифікацію користувачів. Важливою перевагою ГСВЧ є те, що вони генерують випадкові біти, які неможливо передбачити або повторити, що робить їх ідеальними для криптографічних цілей.

8. Визначте послідовність з перших десяти чисел та період лінійного конгруентного ГПСЧ для різних параметрів а, b та c (k0 прийняти рівним - 0):

а) а = 5, b = 7, c = 17;

б) а = 6, b = 3та с = 23.

1. а = 5, b = 7, c = 17;   
   Для визначення послідовності з перших десяти чисел та періоду лінійного конгруентного ГПСЧ для заданих параметрів, ми можемо використати наступну формулу:   
   k[i+1] = (a \* k[i] + b) mod c  
   k[1] = (5 \* 0 + 7) mod 17 = 7  
   k[2] = (5 \* 7 + 7) mod 17 = 8  
   k[3] = (5 \* 8 + 7) mod 17 = 13  
   k[4] = (5 \* 13 + 7) mod 17 = 4  
   k[5] = (5 \* 4 + 7) mod 17 = 10  
   k[6] = (5 \* 10 + 7) mod 17 = 6  
   k[7] = (5 \* 6 + 7) mod 17 = 3  
   k[8] = (5 \* 3 + 7) mod 17 = 5  
   k[9] = (5 \* 5 + 7) mod 17 = 15  
   k[10] = (5 \* 15 + 7) mod 17 = 14   
   Послідовність: (7, 8, 13, 4, 10, 6, 3, 5, 15, 14)   
   Послідовність періодична і має період 10.
2. а = 6, b = 3, с = 23.  
   k[1] = (6 \* 0 + 3) mod 23 = 3  
   k[2] = (6 \* 3 + 3) mod 23 = 21  
   k[3] = (6 \* 21 + 3) mod 23 = 14  
   k[4] = (6 \* 14 + 3) mod 23 = 18  
   k[5] = (6 \* 18 + 3) mod 23 = 19  
   k[6] = (6 \* 19 + 3) mod 23 = 2  
   k[7] = (6 \* 2 + 3) mod 23 = 15  
   k[8] = (6 \* 15 + 3) mod 23 = 1  
   k[9] = (6 \* 1 + 3) mod 23 = 9  
   k[10] = (6 \* 9 + 3) mod 23 = 11  
   Послідовність: (3, 21, 14, 18, 19, 2, 15, 1, 9, 11)  
   Послідовність періодична і має період 10.

9. Визначте послідовність з десяти чисел, що генерується методом Фібоначчі із затримкою, починаючи з таких вихідних даних:

а) a = 3, b = 1, k0 =0,6; k1 =0,3; k2=0,5;

б) a = 4, b = 2, k0 =0,9; k1 =0,3; k2=0,5; k3=0,9.

k3 = k0 - k2 = 0,6 - 0,5 = 0,1;

k4 = k1 - k3 = 0,3 - 0,1 = 0,2;

k5 = k2 - k4 = 0,5 - 0,2 = 0,3;

k6 = k3 - k5 = 0,1 - 0,3 + 1 = 0,8;

k7 = k4 - k6 = 0,2 - 0,8 + 1 = 0,4;

k8 = k5 - k7 = 0,3 - 0,4 + 1 = 0,9;

k9 = k6 - k8 = 0,8 - 0,9 + 1 = 0,9;

k10 = k7 - k9 = 0,4 - 0,9 +1 = 0,5;

k11 = k8 - k10 = 0,9 - 0,5 = 0,4;

k12 = k9 - k11 = 0,9 - 0,4 = 0,5;

Послідовність: (0,1; 0,2; 0,3; 0,8; 0,4; 0,9; 0,9; 0,5; 0,4; 0,5)

б) a = 4, b = 2, k0 =0,9; k1=0,3; k2=0,5; k3=0,9.

k4 = k0 - k2 = 0,9 - 0,5 = 0,4;

k5 = k1 - k3 = 0,3 - 0,9 + 1 = 0,4;

k6 = k2 - k4 = 0,5 - 0,4 = 0,1;

k7 = k3 - k5 = 0,9 - 0,4 = 0,5;

k8 = k4 - k6 = 0,4 - 0,1 = 0,3;

k9 = k5 - k7 = 0,4 - 0,5 + 1 = 0,9;

k10 = k6 – k8 = 0,1 - 0,3 +1 = 0,8;

k11 = k7 - k9 = 0,5 - 0,9 + 1 = 0,6;

k12 = k8 - k10 = 0,3 - 0,8 +1 = 0,5;

k13 = k9 - k11 = 0,9 - 0,6 = 0,3;

Послідовність: (0,4; 0,4; 0,1; 0,5; 0,3; 0,9; 0,8; 0,6; 0,5; 0,3)

10. Значення k0, k1, k2, k3, отримані за допомогою лінійного конгруентного генератора, дорівнюють: k0 = 1, k1 = 8, k2 = 10, k3 = 9. Визначити параметри a, b та c ГПСЧ.

Для визначення параметрів ГПСЧ за значеннями k0, k1, k2 та k3 необхідно виконати наступні кроки:

1. Знайти модуль m як найбільше спільне кратне для чисел k1, k2 та k3. У даному випадку:

m = НСД(8, 10, 9) = 360

2. Знайти одне з можливих значень параметра a, що задовольняє умові:

a ≡ 1 (mod 4)

Одне з можливих значень: a = 13

3. Знайти значення параметра b за формулою:

b ≡ k2 - k1 \* a^(-1) (mod m)

де a^(-1) - мультиплікативно обернений елемент до a по модулю m.

Знайдемо a^(-1) за допомогою розширеного алгоритму Евкліда:

13 \* x ≡ 1 (mod 360)

Розв'язуючи це рівняння, отримаємо: x = 53

Тоді:

b ≡ 10 - 8 \* 13^(-1) (mod 360)

b ≡ 10 - 8 \* 53 ≡ 334 (mod 360)

Отже, b = 334

4. Знайти параметр c за формулою:

c ≡ k3 - a \* b (mod m)

c ≡ 9 - 13 \* 334 ≡ 315 (mod 360)

Отже, c = 315

Отже, параметри ГПСЧ: a = 13, b = 334, c = 315.

11. Обчислити х11 методом генерації ПСЧ BBS, якщо:

а) p = 19, q = 23, х = 3;

M = p \* q = 19 \* 23 = 437

x11 = 3^(2^11 mod((19-1)(23-1)) mod 437 = 24

б) p = 23, q = 31, х = 3.

M = p \* q = 23 \* 31 = 713

x11 = 3^(2^11 mod((23-1)(31-1)) mod 713 = 231

12. Обчислити псевдовипадкову двійкову послідовність довжиною 12 біт методом генерації ПСЧ BBS, якщо:

Як випадкові біти брати молодший біт у двійковому записі числа починаючи з x0.

а) p = 19, q = 23, х = 3;

Обчислимо параметри M та x1:

M = p \* q = 19 \* 23 = 437

x1 = x^2 mod M = 3^2 mod 437 = 9

Для генерації бітів використаємо формулу: xi = (xi-1)^2 mod M. Починаючи з x1, маємо:

x2 = (x1)^2 mod M = 81 mod 437 = 81

x3 = (x2)^2 mod M = 6561 mod 437 = 6

x4 = (x3)^2 mod M = 36 mod 437 = 36

x5 = (x4)^2 mod M = 1296 mod 437 = 422

x6 = (x5)^2 mod M = 178084 mod 437 = 225

x7 = (x6)^2 mod M = 50625 mod 437 = 370

x8 = (x7)^2 mod M = 136900 mod 437 = 119

x9 = (x8)^2 mod M = 14161 mod 437 = 177

x10 = (x9)^2 mod M = 31329 mod 437 = 302

x11 = (x10)^2 mod M = 91204 mod 437 = 308

x12 = (x11)^2 mod M = 94864 mod 437 = 35

Послідовність бітів: 110110011001

б) p = 23, q = 31, х = 3.

Обчислимо параметри M та x1:

M = p \* q = 23 \* 31 = 713

x1 = x^2 mod M = 3^2 mod 713 = 9

Для генерації бітів використаємо формулу: xi = (xi-1)^2 mod M. Починаючи з x1, маємо:

x2 = (x1)^2 mod M = 81 mod 713 = 81

x3 = (x2)^2 mod M = 6561 mod 713 = 144

x4 = (x3)^2 mod M = 20736 mod 713 = 59

x5 = (x4)^2 mod M = 3481 mod 713 = 629

x6 = (x5)^2 mod M = 395641 mod 713 = 639

x7 = (x6)^2 mod M = 408321 mod 713 = 485

x8 = (x7)^2 mod M = 235225 mod 713 = 648

x9 = (x8)^2 mod M = 419904 mod 713 = 660

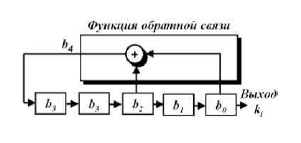
x10 = (x9)^2 mod M = 435600 mod 713 = 670

x11 = (x10)^2 mod M = 448900 mod 713 = 423

x12 = (x11)^2 mod M = 178929 mod 713 = 679

Послідовність бітів: 110111100011.

13. Визначте ключовий потік на 20 біт, який буде згенерований представленим LFSR:



якщо початкове значення послідовності – (1110) 2.

Для визначення ключового потоку на 20 біт, який буде згенерований LFSR з початковим значенням 1110, потрібно виконати наступні кроки:

* Початкове значення регістру LFSR: 1110.
* Згенерувати 20 біт ключового потоку.
* Перший біт ключового потоку визначається як найменший біт початкового значення регістру LFSR: b0 = 0 (1+1+1=3 - непарне число, тому b0 = 1).
* Пересунути біти регістру на один вліво, зберігаючи біт b0 на місці найстаршого біту: 1101.
* Визначити другий біт ключового потоку як найменший біт оновленого значення регістру: b1 = 1 (1+0+1=2 - парне число, тому b1 = 0).
* Пересунути біти регістру на один вліво і додати b1 на місце найстаршого біту: 1010.
* Повторити кроки 5 і 6 для визначення наступних бітів ключового потоку:

b2 = 1, LFSR = 0101

b3 = 1, LFSR = 1010

b4 = 0, LFSR = 1101

b5 = 1, LFSR = 1110

b6 = 0, LFSR = 0111

b7 = 1, LFSR = 1011

b8 = 0, LFSR = 1101

b9 = 1, LFSR = 1110

b10 = 0, LFSR = 0111

b11 = 0, LFSR = 0011

b12 = 1, LFSR = 1001

b13 = 1, LFSR = 1100

b14 = 0, LFSR = 0110

b15 = 0, LFSR = 0011

b16 = 1, LFSR = 1001

b17 = 0, LFSR = 0100

b18 = 0, LFSR = 0010

b19 = 1, LFSR = 1001

b20 = 0, LFSR = 0100

Таким чином, ключовий потік на 20 біт, що буде згенерований LFSR з початковим значенням 1110, буде мати вигляд: 10110011011101101000

14. Для потокового алгоритму A5/1 знайдіть значення функцій: а) f(x, y, z) = f(1, 0, 0); б) f(x, y, z) = f(0, 1, 1); в) f(x, y, z) = f(0, 0, 0); г) f(x, y, z) = f(1, 1, 1). У кожному випадку показати, скільки синхронізується лінійних регістрів зсуву.

У кожному випадку показати, скільки синхронізується лінійних регістрів зсуву.

Алгоритм A5/1 використовує три лінійних регістри зсуву з такими параметрами:

* Регістр 1 (19 біт): початкове значення x = (1001010100000011000)
* Регістр 2 (22 біти): початкове значення y=(0101100110001000101000)
* Регістр 3 (23 біти): початкове значення z = (11001000100001110000000)

Для генерації ключового потоку з кожного регістра витягуються біти зі старшого кінця, використовуючи многочлени зворотнього зв'язку:

* Регістр 1: x^19 + x^18 + x^17 + x^14 + 1
* Регістр 2: y^22 + y^21 + 1
* Регістр 3: z^23 + z^22 + x^21 + z^8 + 1

Для обчислення значення функції f(x, y, z) необхідно виконати побітову операцію XOR над відповідними бітами з трьох регістрів зсуву. Зауважимо, що значення функції залежить тільки від трьох бітів, тому лінійні регістри зсуву синхронізуються не після кожної ітерації алгоритму, а після кожних 64 кроків.

a) f(1, 0, 0) = x[18] XOR y[21] XOR z[22] = 1 XOR 0 XOR 0 = 1

Синхронізація: 64 кроки

б) f(0, 1, 1) = x[19] XOR y[22] XOR z[23] = 0 XOR 0 XOR 1 = 1

Синхронізація: 64 кроки

в) f(0, 0, 0) = x[18] XOR y[21] XOR z[22] = 1 XOR 0 XOR 0 = 1

Синхронізація: 64 кроки

г) f(1, 1, 1) = x[18] XOR y[21] XOR z[22] = 1 XOR 1 XOR 1 = 1

Синхронізація: 64 кроки