***Лабораторна робота №2 Гамування. Моделювання роботи скремблера***

***Мета роботи***

Освоїти практично застосування режиму одноразового гамування. Дослідити побітове неперервне шифрування даних. Ознайомитись із шифруванням інформації за допомогою скремблера.

***Гамування***

Найпростішою і в той же час найбільш надійною з усіх схем шифрування є так звана схема одноразового (однократного) використання (рис. 1), винахід, який найчастіше пов'язують з ім'ям Г.С. Вернама.

*Гама шифру - псевдовипадкова послідовність, створена згідно з заданим алгоритмом для зашифрування відкритих даних і розшифрування зашифрованих даних.*

Гамування – це накладання за певним законом на відкриті дані (дані, що підлягають зашифровуванню) криптографічної гами, тобто, послідовності елементів даних, що виробляються за допомогою деякого криптографічного алгоритму, для отримання зашифрованих даних.

З точки зору теорії криптоаналізу, метод шифрування випадковою одноразовою рівноймовірною гамою тієї ж довжини, що і відкритий текст, є нерозкривним. Крім того, навіть розкривши частину повідомлення, дешифрувальник не зможе хоч скільки-небудь виправити положення - інформація про розкриту ділянку гами не дає інформації про інші її частини.

Процес зашифрування полягає в генерації гами шифру і накладанні отриманої гами на вихідний відкритий текст зворотним чином, наприклад, з використанням операції додавання за модулем 2.

Слід зазначити, що перед зашифруванням відкриті дані  розбивають на блоки однакової довжини, як правило, по 64 бітів. Гама шифру генерується у вигляді послідовності блоків  аналогічної довжини.

Рівняння зашифрування можна записати у вигляді

,

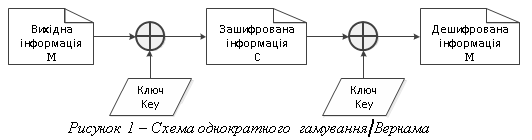
де

1≤ i ≤ *m*, *m* – кількість блоків відкритого тексту;

 – *i*-й блок шифротексту;

 – *i*-й блок гами шифру;

 – *i*-й блок відкритого тексту.



Процес розшифрування зводиться до повторного накладенню тієї ж (або нової) гами на зашифровані дані. Рівняння розшифрування має вигляд

.

По суті гама шифру повинна випадково змінюватися для кожного блоку, що зашифровується. Якщо період гами перевищує довжину всього тексту, що зашифровується, і зловмисникові невідома жодна частина вихідного тексту, то такий шифр можна розкрити тільки прямим перебором усіх варіантів ключа. У цьому випадку криптостійкість шифру визначається довжиною ключа.

***1. Шифр звичайного накладання двійкової гами***

Процес шифрування звичайним накладанням двійкової гами полягає у

наступному:

1. Відкритий текст подають у вигляді неперервної послідовності   
k-розрядних двійкових чисел. Для цього використовують перетворення знаків відкритого тексту із алфавіту обсягом m=2k.

2. Генерують гаму шифру у вигляді послідовності псевдовипадкових двійкових цифр.

3. На кожний черговий розряд відкритого тексту накладають відповідний розряд двійкової гами з використанням операції додавання по модулю 2 (виключна диз'юнкція, або операція XOR, додавання за модулем два – двомісна логічна операція, що приймає значення «істина» тоді і тільки тоді коли значення «істина» має рівно один з її операндів)і таким шляхом отримують черговий двійковий розряд криптограми.

4. Подають криптограму через даний алфавіт, виконавши попереднє розбиття криптограми на послідовні к-розрядні двійкові числа.

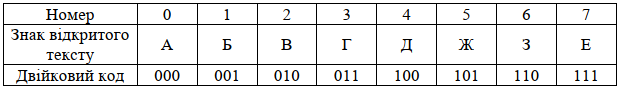
Рівність m=2k є природнім обмеженням даного методу. Якщо m>2k, то у вигляді k-розрядних двійкових чисел неможливо представити всі знаки алфавіту. Якщо m<2k, то не кожне двійкове число може бути представлене у вигляді знаку даного алфавіту. В останньому випадку залишається тільки відмовитися від цієї операції.

Ключові дані, що входять до складу шифру, такі:

– алфавіт з порядковими номерами знаків, а також кількість розрядів двійкового їх подання;

– параметри обраного методу генерації двійкової гами шифру, в тому числі кількість розрядів двійкового подання кожного псевдовипадкового числа їх послідовності.

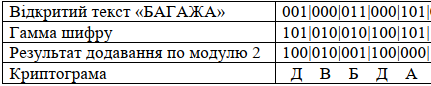
Розглянемо приклади шифрування відкритих текстів, побудованих в алфавіті обсягом m=8 знаки цього алфавіту можна представити у вигляді   
3-розрядних двійкових чисел:



Двійкову гамму шифру згенеруємо лінійним конгруентним методом хn+1=(a xn+b) mod m, n =0, 1, 2,… з параметрами m=16, a=5, b=7, x0=10.

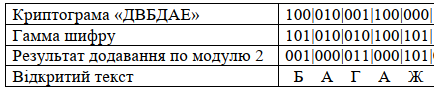
Початок послідовності псевдовипадкових чисел має вигляд: 10, 9, 4, 11, 14, …. Оскільки тут m=24, то кожне десяткове число слід представляти не менше як чотирирозрядне двійкове. У випадку 4-х розрядного подання, початок двійкової гами такий: 1010|1001|0100|1011|1110 ….

***Шифрування:***



При дешифрування криптограми на неї накладається та ж сама двійкова гамма.

***Дешифрування:***



Розглянемо приклад. Для ілюстрації шифрування тексту методом гамування візьмемо двійкові коди символів відкритого тексту *M* та як гаму *G* – випадкову послідовність двійкових кодів.

Нехай маємо відкритий текст M=«SMAIL».

Поставимо у відповідність кожному символу відкритого тексту його ASCII код, а потім переведемо у двійкову систему числення

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Відкритий текст | S | M | A | I | L |
| ASCII -код | 83 | 77 | 65 | 73 | 76 |
| Двійковий код | 1010011 | 1001101 | 1000001 | 1001001 | 1001100 |

Сформуємо випадкову послідовність бітів такої ж довжини, що й двійкове представлення відкритого тексту – гаму. Наприклад G={1010011101010010001111010111}.

Виконуємо шифрування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Відкритий текст | 1010011 | 1001101 | 1000001 | 1001001 | 1001100 |
| Гама | 1100001 | 1010011 | 1010100 | 1000111 | 1010111 |
| Шифротекст | 0110010 | 0011110 | 0010101 | 0001110 | 0011011 |
| ASCII-код | 50 | 94 | 107 | 108 | 89 |
| Шифротекст | 2 | ^ | K | l | Y |

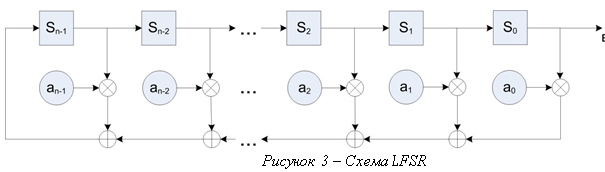
У результаті одержали шифротекст C=«2^klY», що відповідає відкритому тексту M=«SMAIL».

*2 Скремблер*

Скремблером називається програмна чи апаратна реалізація алгоритму, що дозволяє шифрувати побитно неперервні потоки інформації.

Розглянемо зсувний регістр зі зворотним зв'язком (LFSR – Linear Feedback Shift Register) - логічний пристрій, схема на рис.2.

Зсувний регістр є послідовністю *n* біт (*n*-бітний зсувний регістр). Щоразу, коли потрібно отримати біт, всі біти зсувного регістру пересуваються вправо на 1 позицію. Новий крайній лівий біт є функцією всіх інших бітів реєстру. На виході зсувного регістру виявляється молодший біт.

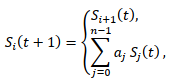


LFSR складається з осередків пам'яті, двійкові стани яких у моменти часу t=0,1,2,…характеризуються значеннями .. Виходи комірок пам'яті пов'язані не тільки послідовно один з одним, але і з суматорами  відповідно до коефіцієнтів передачі : якщо , то значення -ї комірки передається на один із входів i-го суматора; якщо ж , то така передача відсутня. Зазвичай коефіцієнти передачі задаються за допомогою полінома:



Стан LFSR в даний час задається двійковим n-вектор-стовпцем

 Вміст комірок LFSR з часом змінюється так, визначаючи цим динаміку станів LFSR:

При досить довгій роботі скремблера неминуче виникає його зациклювання. Після виконання певного числа тактів в комірках скремблера створиться комбінація біт, яка в ньому вже одного разу виявлялася, і з цього моменту шифруюча послідовність почне циклічно повторюватися з фіксованим періодом.

Щоб побудувати N-розрядний скремблер, що створює послідовність найбільшої довжини, користуються примітивними многочленами. Примітивний (базовий) многочлен ступеня по модулю 2 – це незвідний многочлен, який є дільником , але не є дільником  для всіх d, на які ділиться . Незведений многочлен ступеня *n* не можна у вигляді добутку інших многочленів, крім нього самого та одиничного.

Знайдений примітивний многочлен ступеня *n* записується в двійковому вигляді, потім відкидається одиниця, що відповідає найстаршому розряду.

Наведемо приклад 7-розрядного скремблера, що генерує послідовність з рівним періодом : . Нехай початкове значення стану дорівнюватиме .

Для цього зсувного регістру новий біт генерується за такою схемою:

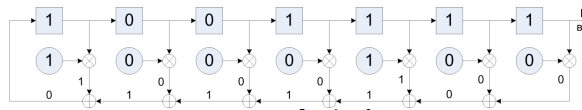


Рисунок 4 – Схема LFSR для многочлена  при початковому стані .

Послідовність зміни внутрішнього стану скремблер має вигляд:

1001111→0100111→1010011→1101001→1110100→0111010→0011101→1001110→0100111→1010011 ...

Рисунок 5 – Зміна стану скремблера

*Завдання до лабораторної роботи № 2*

I. Реалізувати програму для шифрування, що дозволяє виконувати наступні дії:

1. Шифрувати дані в режимі одноразового гамування:

1) текст, що шифрується, повинен зберігатися у файлі;

2) ключ шифрування повинен задаватися випадковим чином;

3) зашифрований текст повинен зберігатися в один файл, а ключ, що використовувався при шифруванні, - в інший;

4) у процесі шифрування передбачити можливість перегляду та зміни ключа, шифрованого та зашифрованого текстів у двійковому, шістнадцятковому та символьному вигляді.

2. Шифрувати дані за допомогою кожного заданого у варіанті скремблера:

1) текст, що шифрується, повинен зберігатися в одному файлі, початкове значення скремблера - в іншому;

2) зашифрований текст повинен зберігатися у файл;

3) у процесі шифрування передбачити можливість перегляду та зміни початкового значення скремблера, шифрованого та зашифрованого текстів у двійковому, шістнадцятковому та символьному вигляді.

ІІ. Реалізувати програму для дешифрування, що дозволяє виконувати такі дії:

1.Дешифрувати дані в режимі одноразового гамування:

1) зашифрований текст повинен зберігатися в одному файлі, ключ – в іншому;

2) розшифрований текст повинен зберігатись у файл;

3) у процесі дешифрування передбачити можливість перегляду та зміни ключа, зашифрованого та розшифрованого текстів у двійковому, шістнадцятковому та символьному вигляді.

2. Розшифрувати дані за допомогою кожного заданого у варіанті скремблера:

1) зашифрований текст повинен зберігатися в одному файлі, початкове значення скремблера – в іншому;

2) зашифрований текст повинен зберігатися у файл;

3) у процесі дешифрування передбачити можливість перегляду та зміни початкового значення скремблера, зашифрованого та розшифрованого текстів у двійковому, шістнадцятковому та символьному вигляді.

ІІІ. За допомогою реалізованих протестувати правильність роботи розроблених програм на наступних задачах:

*1) М =* «*НІЖНО ВПЛІТАЄТЬСЯ В ГОМІН ДНІПРА ДОБРЕ І ЩИРЕ ШЕВЧЕНКІВСЬЕ СЛОВО*»

*Key=*«*СКОМАРОВСЬКИЙ*».

2) Виконати розшифрування шифротексту *С* (ключове слово – «EMPIRE»). При розшифруванні врахувати, що алфавіт містить пробіл, за яким ідують символи латинського алфавіту

*C = «MRPPI FGOUM RYMAH NRYMD UNRWZ OANJE FTIZNI MIQWR EQUNG EIALW RNSMX RJEUA STWYE NCMRY MRCIZ JEEIC XKJQP QXEBD LBJYM LRKME TGJJX EEDIK MFFPB ZJEZD WWCEI DCCIE DBRKF YAIFZ Y».*