Міністерство освіти і науки України
Національний технічний університет України
Київський політехнічний інститут імені Ігоря Сікорського
Фізико-Технічний інститут

Симетрична криптографія

Комп'ютерний практикум 4

Побудова генератора псевдовипадкових послідовностей на лінійних регістрах зсуву (генератора Джиффі) та його кореляційний криптоаналіз

Варіант 4

Виконав:
Волинець Сергій ФІ-03

Київ — 2023

1 Мета роботи

Ознайомлення з деякими принципами побудови криптосистем на лінійних регістрах зсуву; практичне освоєння програмної реалізації лінійних регістрів зсуву (ЛРЗ); ознайомлення з методом кореляційного аналізу криптосистем на прикладі генератора Джиффі.

2 Завдання

2.1 Порядок виконання роботи

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

1. За даними характеристичними многочленами написати програму роботи ЛРЗ L_1, L_2, L_3 і побудованого на них генератора Джиффі.

2. За допомогою формул (4) – (6) при заданому α визначити кількість знаків вихідної послідовності N^*, необхідну для знаходження вірного початкового заповнення, а також поріг для регістрів L_1 та L_2.

3. Організувати перебір всіх можливих початкових заповнень L_1 і обчислення відповідних статистик R з використанням заданої послідовності $(s_i), i = \overline{0, N^* - 1}$.

4. Відбракувати випробувані варіанти за критерієм $R > C$ і знайти всі кандидати на істинне початкове заповнення L_1.

5. Аналогічним чином знайти кандидатів на початкове заповнення L_2.

6. Організувати перебір всіх початкових заповнень L_3 та генерацію відповідних послідовностей (s_i).

7. Відбракувати невірні початкові заповнення L_3 за тактами, на яких $x_i \neq y_i$, де (x_i), (y_i) – послідовності, що генеруються регістрами L_1 та L_2 при знайдених початкових заповненнях.

8. Перевірити знайдені початкові заповнення ЛРЗ L_1, L_2, L_3 шляхом співставлення згенерованої послідовності (z_i) із заданою при $i = \overline{0, N - 1}$.

2.2 Результати

Для виконання даної лабораторної лоботи я вибрав варіант задачі зі спрощеним варіантом. Це пов'язане з тим, що мені потрібно зробити, на мою думку, занадто великий перебір $2^{25} + 2^{26} + 2^{27}$ векторів що складаються з 25, 26 чи 27 бітів.

Отож, для початку, з наступної формули, ми знаходимо значення β.

$$\beta < \frac{1}{2^n}$$

Для L_1: $\beta_1 < 2.98 * 10^{-8}$, для L_2: $\beta_2 < 1.49 * 10^{-8}$.

Тоді, табличні значення кванторів:

$$t_{1-\alpha} = t_{0.99} = 2.32$$
$$t_{1-\beta_1} = t_{0.9999999702} = 5.49$$
$$t_{1-\beta_2} = t_{0.9999999851} = 5.61$$

Після цього за допомогою формул:

$$N \approx \left(\frac{t_{1-\alpha}\sqrt{p_1(1-p_1)} + t_{1-\beta}\sqrt{p_2(1-p_2)}}{p_1 - p_1} \right)^2$$

та

$$C = Np_1 + t_{1-\alpha}\sqrt{Np_1(1-p_1)}$$

де $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{2}$, знаходимо наступні значення:

$$N_1 \geqslant 226, \quad C_1 \approx 71$$
$$N_2 \geqslant 233, \quad C_2 \approx 73$$

Після цього, запустивши написаний алгоритм, я отримав наступні проміжні результати для кандидатів на ключі.

Для L_1, я отримав близько 180 кандидатів. Наприклад:

[000100001101111010110000 1]
[000100010100111010111001 0]
[000100011101111010110000 1]
[000100011101111010110001 1]
[000100011101111010111000 1]
...
[110100100110001000001011 0]
[110100101111001000000010 0]
[111010110001010110011100 0]

Для L_2, натомість, всього два кандидати:

[000000111000100011000011 01]
[100010110111101011000010 11]

А ось, ще декілька значень для L_3:

[000100000100011000000011 000]
[000100000100011000000011 000]
[000100000100011000000011 000]
[000100000100011000000011 000]
[000100000100011000000011 000]
[000100000100011000000011 000]
...

Після цього, запустивши алгоритм для знаходження останнього ключа, я отримав такі результати:

$$k_1 = [000110011101111010110000 1]$$
$$k_2 = [00000011100010001100001101]$$
$$k_3 = [10110001110001110000111 1000]$$

Що і є шуканим ключем.

3 Висновки

Шифри, які пропускають на вихід деяку інформацію про вхідні дані, можна атакувати за допомогою статистичного аналізу. Хоч перебір і буде величензий, але це набагато краще ніж повний перебір.