Лабораторна робота № 5.

Система блочного шифрування S-DES

Спрощений DES – це алгоритм шифрування, який має, скоріше, навчальне, ніж практичне значення. За своїми властивостями він подібний до DES, але має значно менше параметрів.

Цей алгоритм приймає на вході 8-бітний блок відкритого тексту та 10-бітний ключ, а на виході генерує 8-бітний блок шифрованого тексту. При розшифруванні на вхід алгоритму подається 8-бітний блок шифротексту і 10-бітний ключ, а на виході генерує 8-бітний блок відкритого тексту.

Алгоритм шифрування передбачає послідовне виконання п'яти операцій: початкової перестановки IP; раундової функції, що складається з перестановок і підстановок; перестановки SW, коли дві половинки блока по 4 біти переставляються місцями; ще одного застосування раундової функції; і, нарешті, перестановки IP^{-1}, оберненої до початкової. Послідовне використання кількох перестановок і підстановок значно ускладнюють криптоаналіз.

Раундова функція приймає на вході не лише блок тексту, а й 8-бітний цикловий підключ, який утворюється з 10-бітного ключа. Блок-схему алгоритму подано на рис. 2. З цього рисунка видно, що, оскільки це симетричний криптоалгоритм, він використовує для шифрування та розшифрування той самий ключ. Тому ключ має бути як на передавальній, так і на приймальній стороні. З цього ключа на певних етапах шифрування та розшифрування генеруються два 8-бітних раундових підключі.

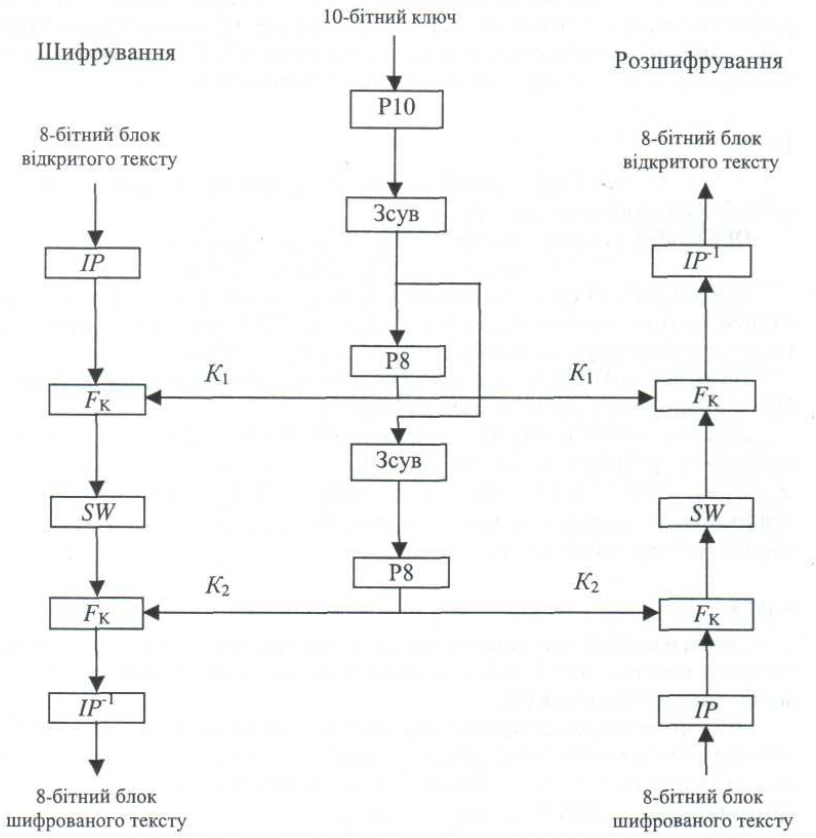

Рис. 2. Схема спрощеного алгоритму S-DES

Процедура генерування раундових підключів

1. Спочатку біти ключа переставляються так. Якщо 10-бітний ключ подати у вигляді $k_1, k_2, \ldots, k_{10}$, то перестановка P10 задається формулою

$$P(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6).$$

Можна також зобразити перестановку P10 у вигляді таблиці:

P10									
3	5	2	7	4	10	1	9	8	6

Ця таблиця символізує позицію біта вхідних даних у вихідній послідовності: першим стає 3-й біт, другим – 5-й, третім – 2-й і т.д. Наприклад, ключ (1010000010) відповідно до цієї перестановки перетворюється в послідовність (1000001100).

2. Ключ розділяється на дві 5-бітні половини. Окремо перша половина й окремо друга піддаються циклічному зсуву ліворуч на одну позицію. У нашому прикладі в результаті буде отримана послідовність (00001 11000).

3. Отримана послідовність піддається перестановці P8, у результаті якої з 10-бітного ключа обирається 8-бітна послідовність за таким правилом:

P8							
6	3	7	4	8	5	10	9

У результаті цієї операції ми отримаємо перший раундовий підключ (K_1). У нашому прикладі він буде мати вигляд (10100100).

4.Для генерування другого раундового підключа K_2 необхідно повернутися на крок назад, до двох 5-бітних рядків до застосування P8 та виконати для кожногоз цих рядків циклічний зсув ліворуч на дві позиції. У нашому прикладі значення підключів (00001 11000) перетворяться у (00100 00011).

5.Нарешті, застосувавши до цієї послідовності перестановку P8, отримаємо другий раундовий підключ K_2. Для нашого прикладу результатом буде (01000011).

Шифрування S-DES

1. *Початкова й кінцева перестановки (*IP *та* IP^{-1}*).* На вхід алгоритму подається 8-бітний блок відкритого тексту, до якого застосовується початкова пеоестановка IP:

IP							
2	6	3	1	4	8	5	7

На завершальній стадії алгоритму виконується обернена перестановка IP^{-1}:

IP^{-1}							
4	1	3	5	7	2	8	6

Можна пересвідчитися, що ці дві таблиці дійсно обернені одна до одної, тобто $IP^{-1}(IP(M)) = M$.

2. *Раундова функція* F_K. Розіб'ємо вхідний блок тексту після IP-перестановки на два 4-бітні підблоки. Лівий 4-бітний блок позначимо L, а правий – R. Тоді циклову функцію можна записати у вигляді формули

$$F_K(L,R) = (L \oplus F(R,K_i),R). \qquad (1)$$

Тут K_i означає цикловий підключ, K_1 або K_2; $\oplus$ – побітне XOR.

Тепер опишемо саму циклову функцію. На вході вона отримує 4-бітне значення (n_1, n_2, n_3, n_4), тобто праву половину вхідного блока. Перша операція – операція розширення та перестановки. Її можна також зобразити таблицею

Розширення з перестановкою							
4	1	2	3	2	3	4	1

Зручніше цю операцію зобразити у вигляді матриці

$$\begin{pmatrix} n_4 & n_1 & n_2 & n_3 \\ n_2 & n_3 & n_4 & n_1 \end{pmatrix}.$$

До цього значення додається 8-бітний підключ за допомогою операції XOR. Це можна зобразити так:

$$\begin{pmatrix} n_4 + k_1 & n_1 + k_2 & n_2 + k_3 & n_3 + k_4 \\ n_2 + k_5 & n_3 + k_6 & n_4 + k_7 & n_1 + k_8 \end{pmatrix}.$$

Перейменуємо отримані елементи:

$$\begin{pmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \end{pmatrix}.$$

Перші чотири біти (тобто перший рядок цієї матриці) далі подаються на вхід модуля заміни (S-матриці), S_0, на виході якого отримується 2-бітна послідовність. Другий рядок матриці подається на вхід другого модуля заміни, S_1, на виході якого також отримується 2-бітна послідовність.

Модулі S_0 і S_1 задаються так:

$$S_0 = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 1 \end{pmatrix} \qquad S_1 = \begin{pmatrix} 1 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix}.$$

Рядки та стовпчики нумеруються, починаючи з нуля.

Ці модулі заміни працюють так. Перший і четвертий біти вхідної послідовності вважаються двійковим представленням номера рядка, а другий і третій – номерами стовпця. Елемент, що знаходиться на перетині цих рядка і

стовпця, задає двобітне вихідне значення. Наприклад, якщо $(p_{00}, p_{03}) = (00)$ та $(p_{01}, p_{02}) = (10)$, то вихідні два біти задаються значенням, яке знаходиться на перетині 0-го рядка та 2-го стовпців, тобто це буде число 3, а у двійковому представленні – 11.

Аналогічну операцію виконують і з другим рядком $p_{10}, p_{11}, p_{12}, p_{13}$.

Після застосування матриць заміни результат піддають перестановці P4 за таким законом:

P4			
2	4	3	1

Результат перестановки P4 і буде результатом функції F_K. Отримана послідовність бітів додається за модулем 2 з лівою половиною L вхідного блока і буде новою лівою половиною. Права половина передається на вихід циклу без змін.

3. *Перестановка підблоків.* Як бачимо, за один раунд раундовою функцією обробляється лише ліва половина відкритого тексту, права половина залишається без змін. Для того, щоби зашифрувати й праву половину, використовується другий цикл, однак на його вхід треба подати переставлені підблоки: L і R поміняти місцями. Для цього й служить функція SW – перемикач блоків. Після переставляння підблоків один раунд алгоритму закінчено. До переставлених підблоків знову застосовується циклова функція, як це описано вище. При другому викликові раундової функції розширення з перестановкою модулі S_0, S_1 і P4 залишаються тими ж, тільки використовується підключ K_2. По закінченні другого раунду виконується IP^{-1} перестановка, й роботу алгоритму закінчено, тобто на виході маємо зашифрований текст.

Розшифрування зашифрованого тексту

Як видно з рис. 2, розшифрування зашифрованого тексту виконується аналогічно шифруванню, за винятком того, що ключі подаються у зворотному порядку.

Завдання роботи.

Використовуючи ключ (0011011101) вручну розшифруйте криптограму свого варіанту, зашифровану алгоритмом S-DES. Продемонструйте проміжні результати, отримані на виході кожної функції (IP, FK, SW, FK, IP^{-1}). Перетворіть отриману двійкову послідовність у десятинне число.

Варіант 1 (11101101)

Варіант 2 (10010110)

Варіант 3 (10110101)

Варіант 4 (01011000)

Варіант 5 (01001000)

Варіант 6 (11010011)

Варіант 7 (00110100)

Варіант 8 (00110110)

Варіант 9 (10100110)

Варіант 10 (11111000)

Варіант 11 (00011010)

Варіант 12 (01110011)

Варіант 13 (11100011)

Варіант 14 (01111101)

Варіант 15 (01011011)

Варіант 16 (01010110)

Варіант 17 (00101011)

Варіант 18 (10010011)

Варіант 19 (10011100)

Варіант 20 (00010011)

Варіант 21 (01001110)

Варіант 22 (11010110)

Варіант 23 (00011001)

Варіант 24 (11010000)

Варіант 25 (00100001)