Лекція № 7 (НЕ 2.1, 2 год.). Модифікації DES. Режими роботи симетричних криптоалгоритмів.

План лекції:

1. Модифікації DES.
2. Режими роботи симетричних криптоалгоритмів.

Зміст лекції:

1. Модифікації DES.

Отже, враховуючи вищезазначене, можна стверджувати, що сьогодні використання DES для критичної, з погляду секретності, інформації є досить небезпечною справою.

Криптологи давно намагалися модифікувати DES із метою збільшення його криптостійкості. Розглянемо такі модифікації, як 3DES та DESX.

Потрійний DES

Найвідомішою модифікацією DES є потрійний DES (3DES), один з варіантів якого визначається формулою:

$$C = 3DES_{K_1K_2K_3}(M) = DES_{K_3}(DES^{-1}_{K_2}(DES_{K_1}(M))).$$

Тут використовуються три ключі K_1, K_2, K_3, сумарна довжина яких складає 56 х 3=186 біт. 64-бітовий блок відкритого повідомлення M спочатку шифрується на ключі K_1, потім розшифровується на ключі K_2 і знову зашифровується на K_3.

Причиною того, що на другому кроці використовується DES^{-1}, а не DES є сумісність з DES. Дійсно, якщо $K_1=K_2=K_3$, то $3DES_K=DES_K$. Причиною того, чому обрано саме три ітерації, а не дві, є існування атаки «зустріч посередині» на подвійний DES.

Розшифрування інформації в 3DES використовується обернено:

$$M = DES^{-1}_{K_3}(DES_{K_2}(DES^{-1}_{K_1}(C))).$$

Отже, кожен блок повідомлення M обробляється DES-машиною тричі на різних ключах, що звичайно, поліпшує криптостійкість. Однак проблемою 3DES є його повільність ñ швидкість його втричі менша за DES. У багатьох випадках це неприпустимо, особливо для шифрування каналів зв'язку.

DESX

У 1984 р. Рон Рівест запропонував інший варіант модифікації DES, позбавлений цього недоліку 3DES. DESX (DES extended - розширений DES) можна визначити так:

$$C = DESX_{KK_1K_2} = K_2 \oplus DES_K(K_1 \oplus M).$$

Отже, 64-бітовий блок повідомлення M підсумовується за mod 2 з першим «зашумлюючим» ключем K1, потім обробляється DES-машиною з ключем K, після чого підсумовується за mod 2 з другим ´зашумлюючимª ключем K2.

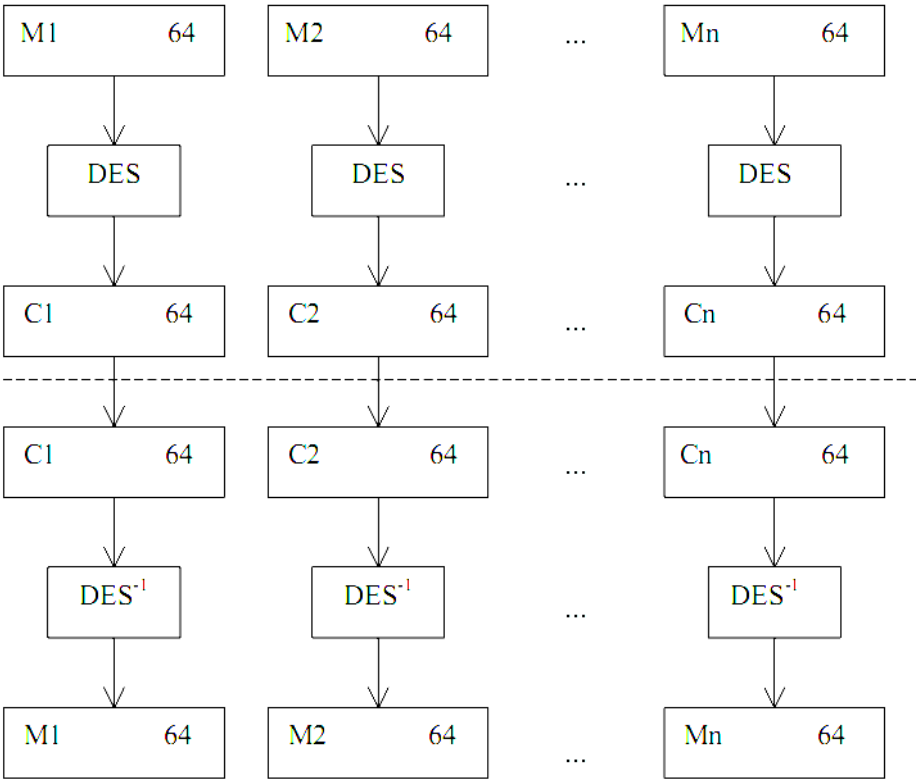
Таким чином, DESX має усього на дві операції XOR більше за оригінальний DES-алгоритм, що не вимагає значних витрат часу.

Водночас наявність цих двох операцій XOR значно поліпшує стійкість до повного перебирання ключів (загальна довжина ключа дорівнює $K_1+K_2+K_3=56+64+64=184$ біти), а також робить його стійкішим до диференційного та лінійного криптоаналізів, збільшуючи необхідну кількість спроб із вибраним текстом до 2^{60}.

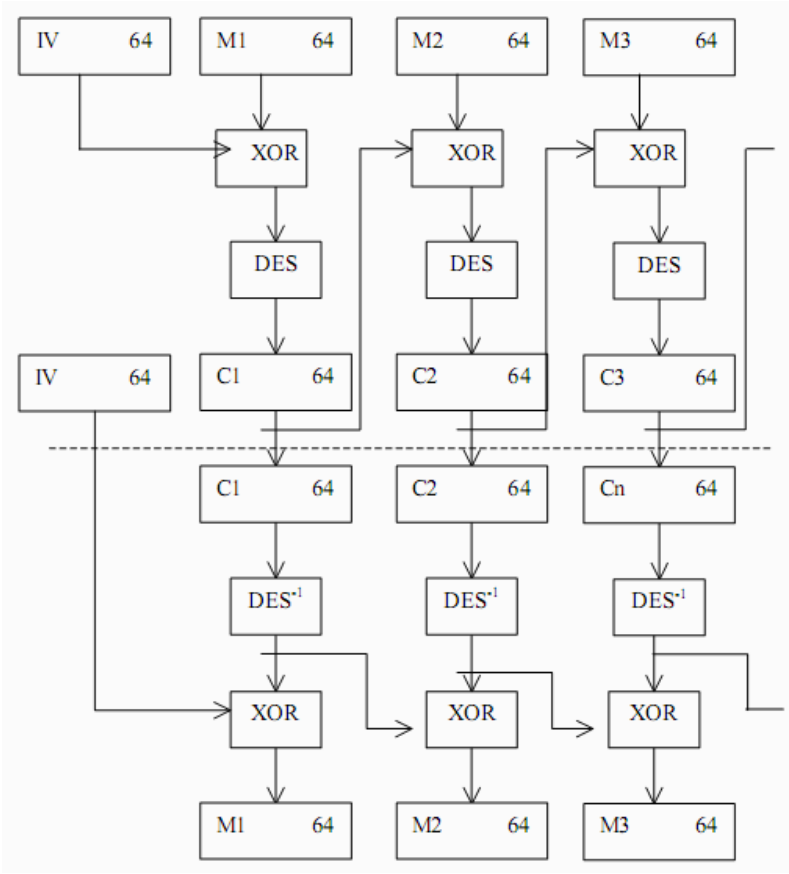
2. Режими роботи симетричних криптоалгоритмів.

Для шифрування вихідного тексту довільної довжини блокові шифри можуть використовуватися в декількох режимах. Існує чотири режими застосування блокових шифрів, які найчастіше зустрічаються в системах криптографічного захисту інформації, а саме режими: електронної кодової книги (ECB ñ Electronic Code Book), зчеплення блоків шифрованого тексту (CBC ñ Cipher Block Chaining), оберненого зв'язку по шифрованому тексту (CFB - Cipher Feedback) і оберненого зв'язку по виходу (OFB - Output Feedback).

У режимі електронної кодової книги кожен блок вихідного тексту шифрується блоковим шифром незалежно від інших. Стійкість цього режиму рівна стійкості самого шифру. Але структура вихідного тексту при цьому не приховується. Кожний однаковий блок вихідного тексту приводить до появи однакового блока шифрованого тексту. Вихідним текстом можна легко маніпулювати шляхом видалення, повторення чи перестановки блоків. Швидкість шифрування дорівнює швидкості блокового шифру. Режим ECB допускає просте розпаралелювання для збільшення швидкості шифрування. Проте жодна обробка неможлива до надходження блока, за винятком генерації ключів.



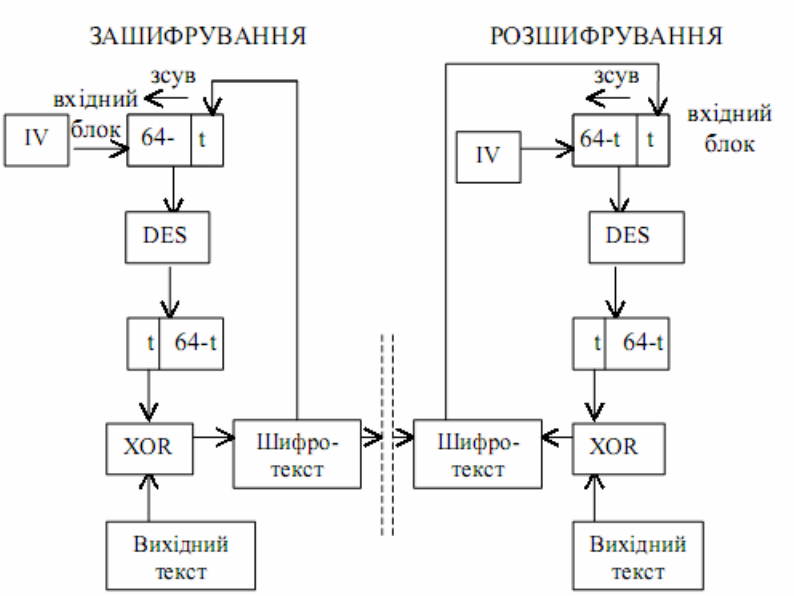
У режимі зчеплення блоків шифрованого тексту (CBC) кожний блок вихідного тексту додається порозрядно за модулем 2 з попереднім блоком шифрованого тексту, а потім шифрується. Для початку процесу шифрування використовується синхропосилка (початковий вектор), яка передається по каналу зв'язку у відкритому вигляді. Стійкість даного режиму рівна стійкості блокового шифру, який лежить у його основі. Крім того, структура вихідного тексту приховується за рахунок додавання попереднього блока шифрованого тексту з черговим блоком відкритого тексту. Стійкість шифрованого тексту збільшується, оскільки стає неможливою пряма маніпуляція вихідним текстом, хіба що шляхом видалення блоків із початку чи кінця шифрованого тексту. Однією з потенційних проблем режиму CBC є можливість внесення контрольованих змін у наступний розшифрований блок вихідного тексту. Наприклад, якщо зловмисник змінить хоча б один біт в блоці, то весь блок буде розшифровано неправильно, але в наступному блоці з'явиться помилка у відповідному положенні. Для боротьби із загрозами вихідний текст повинен містити певну надлишковість.



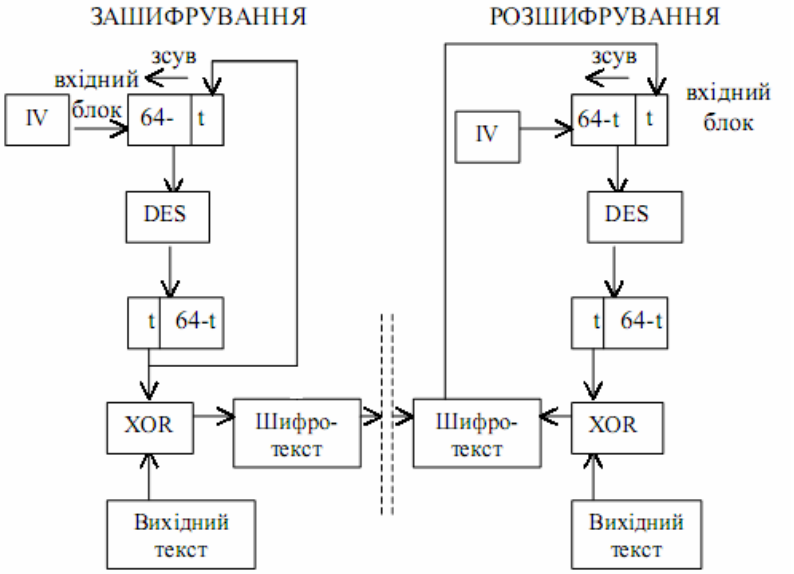
Відомі певні модифікації режиму CBC. Розглянемо деякі з них. Режим зчеплення блоків шифрованого тексту з поширенням (PCBC - Propogating CBC) відрізняється тим, що за модулем 2 додається як попередній блок шифрованого, так і вихідного текстів.

Режим зчеплення блоків шифрованого тексту з контрольною сумою (CBCC - CBC with Checksum) відрізняється тим, що до попереднього блока вихідного тексту перед шифруванням додається сума за модулем 2 всіх попередніх блоків вихідного тексту. Це дає можливість проконтролювати цілісність переданого тексту з невеликими додатковими витратами.

У режимі оберненого зв'язку по шифрованому тексту (CFB) передавальний блок шифрованого тексту шифрується ще раз, і для отримання чергового блока шифрованого тексту результат додається порозрядно за модулем 2 з блоком вихідного тексту. Для початку процесу шифрування також використовується початковий вектор. Стійкість даного режиму рівна стійкості блокового шифру, який лежить у його основі. Структура вихідного тексту приховується за рахунок використання операції додавання за модулем 2. Маніпулювання вихідним текстом шляхом видалення блоків із початку до кінця шифрованого тексту стає неможливим. У режимі CFB, якщо два блоки шифрованого тексту ідентичні, то результати їх шифрування на наступному кроці також будуть ідентичними, а це створює можливість витоку інформації про вихідний текст. Швидкість шифрування дорівнює швидкості роботи блокового шифру і простого способу розпаралелювання процесу шифрування також не здійснюється.



Режим оберненого зв'язку по виходу (OFB) подібний до режиму CFB, за винятком того, що величини, які додаються за модулем 2 з блоками вихідного тексту, генеруються незалежно від вихідного чи шифрованого тексту. Для початку процесу шифрування також використовується початковий вектор. Режим OFB має переваги над режимом CFB, у тому сенсі, що будь-які бітові помилки, які виникають у процесі передачі, не впливають на розшифрування наступних блоків. Проте можлива проста маніпуляція вихідним текстом шляхом зміни шифрованого тексту. Існує модифікація цього режиму, що називається режимом оберненого зв'язку по виходу з нелінійною функцією (OFBNLF - OFB with a NonLinear Function). В цьому випадку на кожному кроці змінюється й ключ шифрування. Хоч у цьому випадку простого способу розпаралелювання процесу шифрування також не існує, час можна зекономити, виробивши ключову послідовність заздалегідь.



У стандарті ГОСТ 28147-89 передбачається використання однойменного алгоритму в режимі генерування імітовставки.

Імітовставкою (або криптографічною контрольною сумою) називають контрольну комбінацію, що додається до повідомлення з метою захисту системи зв'язку від нав'язування хибних даних. Імітовставка залежить від змісту всього повідомлення та ключів шифрування. Наявність імітовставки робить практично нереальним розв'язання двох таких задач без знання ключа:

- обчислення імітовставки для заданої відкритої інформації;
- підбір відкритої інформації під задану імітовставку.

Імітовставка обчислюється для відкритих текстів довжиною від 128 бітів.

Як імітовставку, здебільшого, обирають 32 молодших біти результуючого блока. Вихідний текст M розбивається на блоки M_i по 64 біти. Перший блок M_1 подається на ГОСТ-машину. Отриманий блок шифротексту C_1 додається за модулем 2 з другим блоком M_2 відкритого тексту; результат C_2 - із третім блоком M_3 і т.д. до закінчення відкритого тексту. В результаті на виході отримується один 64-бітовий блок, який залежить від усього повідомлення M та ключа шифрування. Імітовставкою обирають 32 молодших біти цього блока.