УДК 004.056.55

В. А. Лужецький, д. т. н., проф.; О. В. Дмитришин

АЛЬТЕРНАТИВНІ РЕЖИМИ БЛОКОВОГО ШИФРУВАННЯ

Проведено аналіз роботи базових режимів блокового шифрування, розглянуто їх конструктивні особливості. На підставі отриманих результатів розроблено нові схеми роботи режимів блокового шифрування, які дозволяють підвищити криптографічну стійкість процесу шифрування.

Ключові слова: *захист інформації, симетричні блокові шифри, режими блокового шифрування, блокові ключі шифрування, етапи модифікації підключів.*

Вступ

Одним із ефективних методів криптографічного захисту інформації на сьогоднішній день є використання симетричних блокових шифрів. Симетричні шифри – це клас криптографічних алгоритмів, що використовують набори ідентичних примітивних операцій та один і той же самий ключ як для шифрування, так і для розшифрування. Існує два види симетричних шифрів: потокові та блокові. Процес шифрування одного елемента відкритого тексту (символу або одного біта), який фактично зводиться до гамування, виконують потокові симетричні шифри, причому здебільшого кожен елемент відкритого тексту зашифровується незалежно один від одного. У симетричних блокових шифрах, на відміну від потокових, обробці підлягають групи елементів відкритого тексту (блоки даних). Під час такого шифрування кожен блок даних, який обробляється, по-перше, піддається перетворенню декілька раундів, що, в свою чергу, спричинює лавинний ефект. По-друге, кожен елемент блоку даних залежить від всіх елементів цього ж блоку.

Симетричні шифри застосовують для зберігання конфіденційних даних на фізичних носіях та шифрування інформації під час її передавання через комп'ютерні мережі.

Однак, якщо безпосередньо використовувати будь-який симетричний блоковий шифр без застосування додаткових криптографічних перетворень, то у процесі шифрування буде наявний ряд недоліків. Зокрема в таких випадках неможливо приховати структуру інформації, яка захищається за рахунок того, що під час шифрування використовуються блоки фіксованого розміру та один і той же самий секретний ключ. Тому, з метою усунення негативних властивостей процесу шифрування і залежно від галузі, застосовують ряд базових режимів блокового шифрування [1], які стандартизовані Національним інститутом стандартизації та технологій (National Institute of Standards and Technology):

- Electronic Codebook (ECB) – режим електронної кодової книги;
- Cipher Block Chaining (CBC) – режим зчеплення блоків зашифрованого тексту;
- Cipher Feedback (CFB) – режим зворотного зв'язку за зашифрованим текстом;
- Output Feedback (OFB) – режим зворотного зв'язку за виходом;
- Counter Mode (CTR) – режим лічильника.

Окрім вищезазначених режимів розроблено ряд нових режимів шифрування [2 – 8], що можуть бути використані під час захисту інформації, зокрема режим CTR [2] прийняли як доповнення до стандарту «NIST Special Publication 800-38A 2001 Edition – Recommendation for Block Cipher Modes of Operation. Methods and Techniques» [1].

Метою цієї роботи є підвищення стійкості процесу шифрування із використанням симетричних блокових шифрів за рахунок використання нових схем режимів роботи блокового шифрування.

Основними задачами досліджень є:

1. Аналіз базових режимів блокового шифрування, що використовують у схемах симетричного блокового шифрування.

2. Розробка підходу щодо використання нових схем роботи режимів блокового шифрування, який дозволить підвищити криптографічну стійкість процесу шифрування.

Аналіз базових режимів блокового шифрування

1. Режим електронної кодової книги

Першим і найпростішим режимом блокового шифрування є режим електронної кодової книги [1]. Під час використання цього режиму відкритий текст розбивають на n-бітні блоки, і кожен блок зашифровують незалежно від інших. Процес зашифрування і розшифрування описують такими формулами:

$$C_i = E_K(P_i),\ P_i = D_K(C_i), \tag{1}$$

де E, D – відповідно функції зашифрування і розшифрування; P_i, C_i – i-ий n-бітний блок відкритого і зашифрованого тексту відповідно, i=1, 2, 3, …; K – n-бітний секретний ключ шифрування.

Основними перевагами режиму ECB є можливість одночасно шифрувати декілька блоків даних і підтримка самосинхронізації, тобто пошкодження i-го блоку зашифрованого тексту впливає лише на той же самий розшифрований блок. Проте такий процес шифрування зумовлює ряд недоліків:

- однакові блоки відкритого тексту зумовлюють появу однакових блоків зашифрованого тексту;
- перестановка блоків зашифрованого тексту спричинює перестановку відповідних блоків відкритих текстів, що призводить до порушення цілісності інформації;
- неможливо приховати структури інформації, яка підлягає захисту.

Частина базових режимів блокового шифрування, зокрема режим ECB, є вразливими до атаки на основі пар відомих текстів і зашифрованих текстів та до атаки «дня народження». Окрім того, відомим фактом є те, що коли великий обсяг інформації зашифровують із використанням одного і того ж секретного ключа, то в блоках зашифрованого тексту присутня інформація про відкритий текст. У роботі [3] наводиться такий факт.

Якщо блоки відкритого тексту зашифровані на одному й тому ж секретному ключі і приймають випадкову форму, яка задовольняє закон рівномірного розподілу, то в зашифрованому тексті присутній витік інформації про відкритий текст з ймовірністю

$$p_s = 1 - \left(1 - 2^{-n}\right)^{s(s-1)/2}, \tag{2}$$

де $s = 2^{(n+1)/2}$, $p_s \approx 0{,}63$.

Згідно парадоксу «дня народження», для режиму ECB, для s n-бітних блоків зашифрованих текстів C_1, …, C_s існує така пара, що $C_i = C_j$ з ймовірністю p_s. Таким чином, перший з недоліків режиму ECB стає вагомою завадою для використання цього режиму, оскільки зловмисник одразу може довідатись, що $P_i = P_j$.

2. Режим зчеплення блоків зашифрованого тексту

У режимі зчеплення блоків зашифрованого тексту [1], i-ий блок відкритого тексту та (i-1)-ий блок зашифрованого тексту зчіплюють за допомогою операції додавання за модулем два перед виконанням процесу зашифрування, який здійснюється таким чином:

$$C_i = E_K(P_i \oplus C_{i-1}),\ P_i = D_K(C_i) \oplus C_{i-1},\ C_0 = IV, \tag{3}$$

де $\oplus$ – операція додавання за модулем два; IV – вектор ініціалізації.

Перевагами режиму CBC є відсутність недоліків, які притаманні ECB, а саме, підтримки процесу приховування структури відкритого тексту та відсутність можливості перестановки блоків зашифрованого тексту, які досягаються за рахунок того, що кожен наступний блок

зашифрованого тексту залежить від всіх попередніх блоків. Однак це в свою чергу не дозволяє одночасно зашифровувати декілька блоків відкритого тексту.

Окрім вищезазначених переваг варто відзначити, що цей режим дозволяє розпаралелювати процес розшифрування блоків зашифрованого тексту і підтримує самосинхронізацію. Якщо під час передачі або запису даних був пошкоджений i-ий блок зашифрованого тексту, то під час розшифрування лише i-ий та (i+1)-ий блоки відкритого тексту будуть пошкодженими.

Однакові блоки відкритого тексту зумовлюють появу різних блоків зашифрованого тексту, що в свою чергу унеможливлює використання процесу перестановки блоків зашифрованого тексту для модифікації змісту відкритого тексту на відміну від режиму ECB. За рахунок цього режим шифрування CBC також використовують в якості засобу для забезпечення цілісності та захисту інформації від фальсифікації.

Проте для режиму CBC, як і для режиму ECB, також існує s n-бітних блоків зашифрованих текстів $C_1, \ldots, C_t$ таких, що $C_i = C_j$ з ймовірністю p_s[3]. Тобто $P_i \oplus C_{i\text{-}1} = P_j \oplus C_{j\text{-}1} \rightarrow P_i \oplus P_j = C_{i\text{-}1} \oplus C_{j\text{-}1}$. У випадку, якщо у атакуючого є доступ до декількох наборів шифротекстів $C^1, \ldots, C^t$, де кожен набір C^l складається з s_l блоків зашифрованого тексту, то загальний обсяг вибірки, з якою працює зловмисник, дорівнює

$$s = \sum_{l=1}^{t} s_l \ . \tag{4}$$

Таким чином, використовуючи парадокс «дня народження» зловмисник може замінити пару блоків зашифрованих текстів ($C_{i\text{-}1}$, C_i) на ($C_{j\text{-}1}$, C_j) навіть незважаючи на те, що блоки відкритих текстів $P_{i\text{-}1}$ та $P_{j\text{-}1}$ не будуть достовірними, щоб отримати під час розшифрування всі інші блоки відкритих текстів, які будуть еквівалентні оригінальним блокам. Більш того, зловмисник, володіючи декількома наборами зашифрованих текстів, може підмінити цілі групи блоків зашифрованих текстів ($C_{i\text{-}1}, \ldots, C_{i\text{-}w}$) на ($C_{j\text{-}1}, \ldots, C_{j\text{-}w}$), якщо $C_i = C_j$ та $C_{i\text{-}w} = C_{j\text{-}w}$.

Також, як зазначено в роботах [7, 9], атакувати базову функцію шифрування в режимі CBC можна із використанням атаки на основі пар відомого відкритого тексту, тобто вхідне значення, яке подають на функцію шифрування розраховують за допомогою додавання за модулем два поточного блоку відкритого тексту та попереднього блоку зашифрованого тексту, а вихід функції шифрування – це поточний блок зашифрованого тексту.

3. Режим зворотного зв'язку за зашифрованим текстом

У режимі зворотного зв'язку за зашифрованим текстом блоки відкритого тексту додають за модулем два з блоками гамми. Перший блок гамми формують із використанням n-бітного початкового вектора ($1 \le n \le b$), який записують в молодші n-бітів b-бітного вхідного блоку I_1, який разом із секретним ключем K подають на функцію шифрування. n-старших бітів вихідного блоку даних з функції шифрування є гаммою, яку додають за модулем два з блоком відкритого тексту, внаслідок чого отримують блок зашифрованого тексту. j-ий вхідний блок I_j є результатом конкатенації (b-n)-молодших біт попереднього вхідного блоку $I_{j\text{-}1}$, які записують в старші (b-n)-біти b-бітного вхідного блоку I_j, та n-біт блоку зашифрованого тексту $C_{j\text{-}1}$, які записують в молодші n-біти b-бітного вхідного блоку I_j і т. д.

Процес зашифрування блоків відкритого тексту та розшифрування блоків зашифрованого тексту виконують таким чином:

$$I_1 = IV \ , \ I_j = \left(I_{j-1} << n\right) \| \, C_{j-1} \, , \ O_j = E_K\left(I_j\right),$$

$$C_j = P_j \oplus \left(O_j >> \left(b-n\right)\right), \ P_j = C_j \oplus \left(O_j >> \left(b-n\right)\right). \tag{5}$$

де I_j – b-бітний вхідний блок; O_j – b-бітний вихідний блок;

>>, << – операція правостороннього та лівостороннього зсуву відповідно.

До позитивних властивостей режиму CFB відносять можливість приховувати структури відкритого тексту та підтримку самосинхронізації, тобто якщо буде втрачений 1 біт

будь-якого блоку зашифрованого тексту, то під час розшифрування буде пошкоджено ще додатково b/n блоків відкритого тексту. Як і в режимі CBC процес розшифрування може бути розпаралелюваний, а процес зашифрування – ні, оскільки кожен наступний блок зашифрованого тексту залежить від всіх попередніх блоків.

У режимі CFB наявний той же недолік при атаці «дня народження», що і в режимі CBC. Тобто з імовірністю p_s існують такі пари зашифрованих текстів, що $C_i = C_j$. Звідки $P_i \oplus (O_i >> (b\text{-}n)) = P_j \oplus (O_j >> (b\text{-}n)) \rightarrow P_i \oplus P_j = (O_i >> (b\text{-}n)) \oplus (O_j >> (b\text{-}n)) \rightarrow P_i \oplus P_j = C_{i\text{-}1} \oplus C_{j\text{-}1}$.

Зокрема, атака базової функції шифрування в режимі CFB може бути здійснена за допомогою атаки на основі пар відомого відкритого тексту [7]. Причому вхідне значення може бути розраховане на підставі інформації про попередні блоки зашифрованого тексту, а вихід функції шифрування отримують за допомогою додавання за модулем два поточного блоку відкритого тексту та відповідного блоку зашифрованого тексту.

4. Режим зворотного зв'язку за виходом

Режим зворотного зв'язку за виходом є конфіденційним режимом, в якому із початкового вектора IV генерують послідовність n-бітних вихідних блоків O_j, які суммують за модулем два з блоками відкритих текстів P_j, щоб отримати блоки зашифрованих текстів і навпаки.

Процес зашифрування та розшифрування, згідно [1], виконують таким чином:

$$I_1 = IV,\ I_j = O_{j-1},\ O_j = E_K\left(I_j\right),\ C_j = P_j \oplus O_j,\ C_N^* = P_N^* \oplus \left(O_j >> \left(n-u\right)\right),$$

$$P_j = C_j \oplus O_j, P_N^* = C_N^* \oplus \left(O_j >> \left(n-u\right)\right), \qquad (6)$$

де I_j – n-бітний вхідний блок; P^*_N, C^*_N – останні u-бітні блоки відкритого та зашифрованого текстів відповідно.

Якщо останній блок відкритого тексту P^*_N складається з u-бітів, то цей блок відкритого тексту додають за модулем два з u найбільш значущими бітами останнього вихідного блоку. Окрім того початковий вектор IV має бути унікальним для кожного повідомлення, яке шифрують на заданому ключі. Можливі способи генерування початкового вектора IV описано в [1].

Головною перевагою режиму OFB порівняно із попередніми режимами є те, що пошкодження одного біта зашифрованого тексту під час розшифрування вплине на той же самий біт відкритого тексту. Також наявна можливість приховування структури відкритого тексту. Проте режиму OFB притаманні такі недоліки:

- відсутність можливості суміщати в часі процеси зашифрування та розшифрування декількох блоків даних;
- необхідність періодичної повторної синхронізації;
- вразливість до зміни окремих біт зашифрованого тексту.

У режимі OFB, як і в попередніх режимах, функція шифрування може бути взломана за допомогою атаки на основі відомих пар текстів [7]. Тобто, якщо відкритий текст і зашифрований текст є відомими, то процес знаходження вихідних блоків з функції шифрування, а отже і вхідних, є достатньо простим.

5. Режим «лічильника»

Режим CTR подібний до попереднього режиму. Проте в режимі «лічильника» процесу шифрування підлягає не попередній вихід функції шифрування, а значення лічильника T_j, яке збільшують на кожному кроці на деяке постійне число. Процес шифрування описують такими формулами [1]:

$$O_j = E_K\left(T_j\right),\ C_j = P_j \oplus O_j,\ C_N^* = P_N^* \oplus \left(O_j >> \left(n-u\right)\right),$$

$$P_j = C_j \oplus O_j, P_N^* = C_N^* \oplus \left(O_j >> \left(n-u\right)\right). \qquad (7)$$

Позитивними особливостями режиму CTR є те, що [2]:
- j-ий блоку зашифрованого тексту C_j може бути зашифрований випадковим чином;
- наявна можливість розпаралелювання процесу зашифрування та розшифрування;
- структуру інформації, яку захищають, можна приховати.

До негативних властивостей режиму CTR відносять такі недоліки:
- помилка в одному біті зашифрованого тексту вплине лише на відповідний біт відкритого тексту;
- необхідна періодична повторна синхронізація лічильника;
- режим вразливий до зміни окремих біт зашифрованого тексту.

Якщо в даному режимі використовують «слабкий» блоковий шифр, то періодичність лічильника може бути використана для застосування атаки диференційного криптоаналізу [2].

Вище зазначені недоліки базових режимів блокового шифрування та специфіка їх застосувань спричинюють пошук альтернативних режимів шифрування.

На сьогоднішній день схеми альтернативних режимів блокового шифрування будують:
- на зчепленні блоків відкритого та зашифрованого тексту, зокрема такі методи описані в роботах [3], [8], [10] тощо;
- на модифікаціях режиму лічильника [7];
- на модифікації режиму зворотного зв'язку за зашифрованим текстом [11];
- за допомогою матричного шифрування, коли один підблок даних шифрують у складі різних блоків даних [6];
- на базі мережі Фейстеля [4, 5].

Вищезазначені базові та альтернативні режими шифрування виконують зав'язку на рівні блоків відкритого та зашифрованого текстів, які мають як переваги, так і недоліки.

Проте існує й інший підхід до побудови режимів блокового шифрування, який ґрунтується на використанні різних ключів шифрування для кожного блоку даних. Даний підхід був реалізований в режимі зворотного зв'язку за ключем (Key Feedback Mode) [12]. Такий режим подібний до режиму OFB, але зворотний зв'язок організовано не за виходом, а за ключем.

Автори цієї статті пропонують підхід, який дозволяє виконувати зав'язку на рівні процедур розгортання ключів.

Режими зчеплення ключів

Будь-який симетричний блоковий шифр описується двома головними складовими: процедурою розгортання ключів та процедурою шифрування даних. Процедуру розгортання ключів використовують для формування набору підключів, які в подальшому використовують у процедурі шифрування. Процедура шифрування виконує безпосереднє перетворення даних, тобто зашифрування та розшифрування блоків даних. Залежно від функціонального призначення, процедури розгортання ключів діляться на дві групи [13].

Першу групу утворюють процедури розгортання ключів, які формують підключі для послідовностей блоків даних. До другої групи належать процедури розгортання ключів, які формують підключі для раундів шифрування одного блоку даних і включають три основних етапи модифікації ключа [14]: початковий, головний та кінцевий.

Початковий етап модифікації ключа призначений для виконання початкових перетворень над секретним ключем та формування набору вхідних значень для етапу головної обробки ключа. На другому етапі з отриманих даних формують набір підключів для їх подальшого використання на етапі кінцевої модифікації. До перетворень, які виконуються на цьому етапі, висувають такі вимоги:
- процес відновлення ключа шифрування K із заданого підключа повинен бути достатньо складним;
- кожен біт вхідного ключа шифрування повинен впливати на кожен підключ.

На етапі кінцевої модифікації ключа виконують перетворення набору підключів у форму придатну для їх використання, в процедурі шифрування.

Вищезазначені етапи модифікації ключа використовують для отримання набору підключів для раундів шифрування першого блоку даних $\{S\}_1$, який фактично і є першим блоковим ключем, з секретного ключа шифрування K. Авторами цієї статті пропонуються такі режими зав'язки блокових підключів.

1. Ітеративний режим зчеплення підключів. Для формування набору підключів $\{S\}_i$ (i=$\overline{2, N}$, де N – кількість наборів підключів) виконують такі перетворення

$$S_{1,i} = f_t\left(S_{k,i-1}\right),\ S_{j,i} = f_t\left(S_{j-1,i}\right), \tag{8}$$

де $S_{j,i}$ – j-ий m-бітний підключ i-го блокового ключа, $j = \overline{1,k}$, k – кількість підключів у наборі; f_t – функція довільного відображення m-біт в m-біт, у разі виконання операції шифрування t, t={e, d}, e – операція зашифрування, d – операція розшифрування; m – розрядність підключів;

Під час виконання цього режиму для формування першого набору підключів $\{S\}_1$ використовують секретний ключ шифрування K. Для отримання першого підключа i-го блокового ключа k-ий підключ (i-1)-го блокового ключа надсилають на функцію f_t. Щоб сформувати j-ий підключ i-го блокового ключа (j-1)-ий підключ i-го блокового ключа надсилають на функцію f_t і т. д. (рис.1).

Для цього режиму неможливо розпаралелити процес обчислення підключів, оскільки для обчислення j-го підключа використовують значення (j-1)-го підключа та для обчислення i-го блокового ключа використовують (i-1)-ий блоковий ключ.

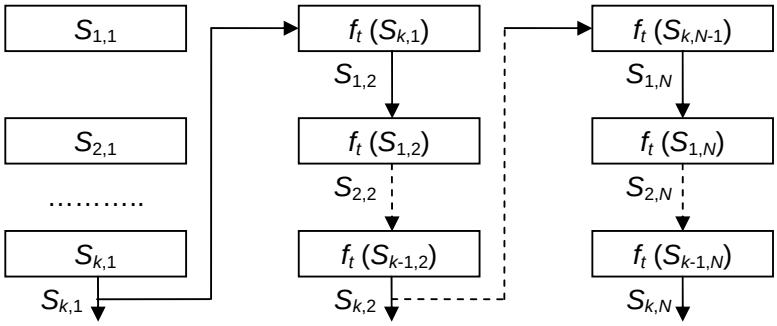

Рис. 1. Схема ітеративного режиму зчеплення підключів

2. Послідовний режим зчеплення підключів використовують для формування набору підключів $\{S\}_i$ за таким правилом

$$S_{j,i} = f_t\left(S_{j,i-1}\right). \tag{9}$$

Як і в попередньому режимі, для формування першого блокового ключа $\{S\}_1$ використовують секретний ключ шифрування K. В отриманому наборі підключів $\{S\}_1$, кожен підключ незалежно один від одного надходить на функцію f_t, з виходу якої отримують наступний блоковий ключ $\{S\}_2$ і т. д. (рис. 2).

Оскільки j-ий та (j-1)-ий підключі i-го блокового ключа не пов'язані один з одним, то для заданого режиму можна організувати процес паралельного обчислення значень підключів.

3. Комбінований режим. Цей режим полягає в одночасному використанні двох попередніх режимів зчеплення і описується такими виразами

$$S_{1,i} = g_t\left(S_{k,i-1}, S_{1,i-1}\right),\ S_{j,i} = g_t\left(S_{j-1,i}, S_{j,i-1}\right), \tag{10}$$

де g_t – функція довільного відображення $2m$-біт в m-біт.

Таким чином, процес отримання другого блокового ключа полягає у виконанні кількох

кроків. По-перше, на функцію g_t надсилають перший $S_{1,i\text{-}1}$ та останній підключі $S_{k,i\text{-}1}$ (i-1)-го блокового ключа, для того, щоб отримати перший підключ i-го блокового ключа. По-друге, отриманий підключ та другий підключ (i-1)-го блокового ключа беруть участь у формуванні другого підключа $S_{2,i}$ другого блокового ключа, які надходять на функцію g_t. Для формування j-го підключа i-го блокового ключа виконують перетворення згідно формули (10) (рис. 3). Перший блоковий ключ обчислюють згідно правил, які задаються в процедурі розгортання ключа шифрування K.

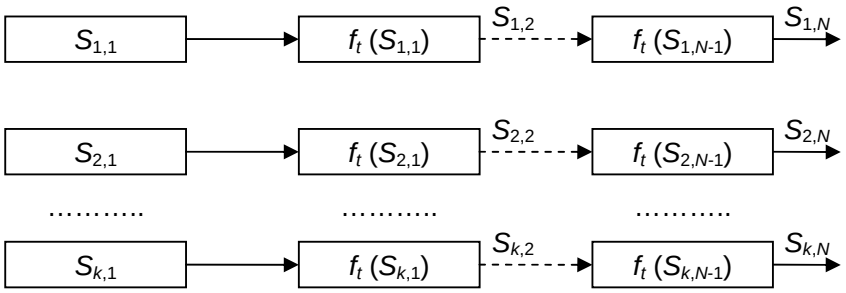

Рис. 2. Схема послідовного режиму зчеплення підключів

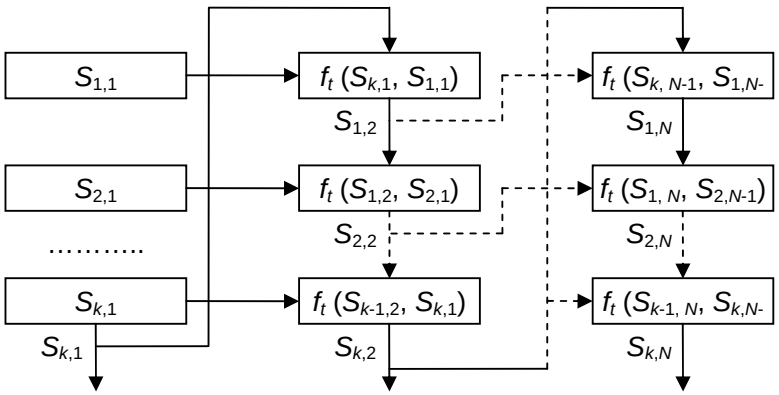

Рис. 3. Схема комбінованого режиму зчеплення підключів

Комбінований режим зчеплення підключів не підтримує процес їхнього розпаралеленого обчислення для i-го блокового ключа, оскільки у формуванні j-го підключа поточного блокового ключа бере участь (j-1)-ий підключ цього ж блокового ключа.

Оскільки в деяких симетричних блокових шифрах [15] час розгортання ключа значно перевищує час зашифрування / розшифрування одного блоку даних, тому для застосування вищерозглянутих режимів зчеплення ключів необхідно дотримуватися такої вимоги: час розгортання підключів повинен бути меншим за час зашифрування та розшифрування одного блоку даних.

Наступним недоліком, який зменшує привабливість застосування режимів зчеплення ключів є збільшення сукупного часу шифрування даних. Навіть якщо час розгортання ключів не перевищуватиме час шифрування даних, то сукупний час приблизно буде збільшено вдвічі. Проте, враховуючи сучасні тенденції розвитку мікропроцесорної техніки, можна зробити висновок, що в найближчому майбутньому кожен персональний комп'ютер буде багатоядерним. Тому виконання процедур розгортання ключів та шифрування даних можна суміщати в часі.

Ключі сформовані в режимі зчеплення ключів забезпечують таке:

- однаковим блокам відкритого тексту відповідають різні блоки зашифрованого тексту;
- можливість приховувати структуру відкритого тексту;

- можливість виявлення факту порушення цілісності повідомлення у разі перестановки блоків тексту;
- процедура шифрування не вимагає векторів ініціалізації;
- можливість одночасного обчислення блокових ключів та виконання процесу шифрування блоків даних;
- відсутність розповсюдження помилок (помилка в одному блоці зашифрованого тексту призводить до помилкового розшифрування тільки цього блоку).

Недоліком використання режиму зчеплення ключів є неможливість одночасно шифрувати декілька блоків даних.

Також, режими зчеплення ключів дозволяють підвищити стійкість блокового шифрування до атаки «дня народження» за рахунок використання блокових ключів. Як наслідок, якщо $C_i = C_j$, то $P_i \neq P_j$. Окрім того, для режимів ECB, CBC та CFB зловмисник, володіючи декількома наборами зашифрованих текстів, міг підмінити цілі групи блоків зашифрованих текстів ($C_{i\text{-}1}$, …, $C_{i\text{-}w}$) на ($C_{j\text{-}1}$, …, $C_{j\text{-}w}$), якщо $C_i = C_j$ та $C_{i\text{-}w} = C_{j\text{-}w}$. У випадку застосування режимів зчеплення ключів така підміна буде неможлива, оскільки відкритий текст після розшифрування буде спотвореним.

Залежно від використовуваного режиму зчеплення ключів період генерування блокових ключів може змінюватися. Цей період буде визначатися функцією формування послідовності ключів, яка повинна:

- бути простою в реалізації, як в програмній, так і в апаратній реалізації, щоб значно не збільшувати час шифрування;
- щоб достатньо складно було відновити (i-1)-ше значення блокового ключа в разі відомого i-го ключа;
- функція має забезпечувати достатньо великий період T ключової послідовності, який дозволяє шифрувати тексти будь-якої довжини.

Висновки

Частина базових режимів блокового шифрування, зокрема режим ECB, CBC і CFB, є вразливими до атаки «дня народження», за допомогою якої під час використання одного і того ж секретного ключа можна отримувати інформацію про відкритий текст та порушувати цілісність відкритого тексту.

Вищезазначені недоліки базових режимів блокового шифрування та специфіка їх застосувань спричинюють пошук альтернативних режимів шифрування, які здебільшого засновані на модифікаціях базових режимів симетричного блокового шифрування та виконують зав'язку на рівні блоків відкритого та зашифрованого текстів.

Проте існує й інший підхід до побудови режимів блокового шифрування, який ґрунтується на використанні різних ключів шифрування для кожного блоку даних, що також підвищує криптографічну стійкість процесу шифрування, зокрема цей підхід був реалізований в режимі зворотного зв'язку за ключем.

У цій роботі запропоновано саме такий підхід, коли для кожного блоку даних використовують різні блокові ключі. Запропоновані режими забезпечують формування власних блокових ключів для кожного блоку даних, що підвищує криптографічну стійкість процесу шифрування, зберігаючи при цьому більшу частину переваг базових режимів шифрування. Таким чином, зловмиснику, наприклад, для здійснення атаки з відновлення одного блокового ключа необхідно опрацювати в T раз більше пар зашифрованих / розшифрованих текстів, ніж для базових режимів шифрування.

СПИСОК ЛІТЕРАТУРИ

1. NIST Special Publication 800-38A 2001 Edition – Recommendation for Block Cipher Modes of Operation. Methods and Techniques // Computer Security Division Information Technology Laboratory National Institute of Standards and Technology, Gaithersburg, MD 20899-8930. December 2001.

2. Lipmaa H. CTR-mode encryption / H. Lipmaa, Ph. Rogaway and D. Wagner // Submission of modes of operation, 2001. – P. 4. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ctr/ctr-spec.pdf.

3. Knudsen L. R. Block chaining modes of operation / L. R. Knudsen // Reports in informatics No 207, October 2000. – P. 16. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/abc/abc-spec.pdf.

4. Brier E. BPS: a format-preserving encryption proposal / E. Brier, Th. Peyrin and J. Stern – P. 11. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/bps/bps-spec.pdf.

5. Bellare1 M. The FFX Mode of Operation for Format-Preserving Encryption. Draft 1.1. / M. Bellare1, Ph. Rogaway and T. Spies // February 20, 2010. – P. 18. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM documents/proposedmodes/ffx/ffx-spec.pdf.

6. Belal A. A. 2D-Encryption mode. / A. A. Belal, M. A. Abdel-Gawad // March, 2001. – P. 32. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/2dem/2dem-spec.pdf.

7. Головашич С. А. Безопасность режимов блочного шифрования / С. А. Головашич // Радиотехника: Всеукр. межвед. научн.-техн. сб. – Х.: ХНУРЭ, 2001. – Вып. 119. – С. 135 – 145.

8. Дмитришин О. В. Режим керованого зчеплення блоків зашифрованого тексту / О. В. Дмитришин, В. А. Лужецький // Вісник ВПІ. – Вінниця, Видавництво Вінницького національного університету, 2009 – № 1. – С. 34 – 36.

9. Biham E. Differential cryptanalysis of Feal and N-hash / E. Biham, A. Shamir // Advances in Cryptology Proceedings Eurocrypt'91, LNCS 547, D.W. Davies, Ed.,Springer-Verlag, 1991. – pp. 1 – 16.

10. Gligor V. D. On Message Integrity in Symmetric Encryption / V. D. Gligor, P. Donescu // November 10, 2000. – P. 41. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ige/ige-spec.pdf.

11. Mattsson Ul. T. Format-controlling encryption using datatype-preserving encryption / Ul. T. Mattsson // June 30, 2009. – P. 46. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/fcem/fcem-spec.pdf.

12. Hastad J. Key Feedback Mode: a Keystream generator with Provable Security / J. Hastad, M. Naslund // October 11, 2000. – P. 23. – Режим доступу до ресурсу: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/kfb/kfb-spec.pdf.

13. Лужецький В. А. Процедури розгортання ключів для блокових шифрів на основі арифметичних операцій за модулем / В. А. Лужецький, О. В. Дмитришин // Інформаційні технології та ком'ютерна інженерія. – Вінниця, Видавництво Вінницького національного університету, 2009 – № 2. – С. 69-74.

14. Коркішко Т. Алгоритми та процесори симетричного блокового шифрування / Коркішко Т., Мельник А., Мельник В. – Львів: Бак, 2003. – 168 с. – ISBN 966-7065-43-X.

15. Горбенко І. Д. Аналіз властивостей алгоритмів блокового симетричного шифрування (за результатами міжнародного проекту NESIE) / І. Д. Горбенко, Г. М. Гулак та інші // Радиотехника: Всеукр. межвед. научн.-техн. сб. – Х.: ХНУРЭ, 2005 – № 141. – С. 7 – 24.

Лужецький Володимир Андрійович – д. т. н., професор, завідуючий кафедри захисту інформації.

Дмитришин Олександр Васильович – магістр з інформаційної безпеки, аспірант кафедри захисту інформації. E-mail: olexanderdm@gmail.com

Вінницький національний технічний університет