УДК 004.056.55

О.О. КУЗНЕЦОВ, Р.І. МОРДВІНОВ, Є.П. КОЛОВАНОВА, А.В. САМОЙЛОВА

ДОСЛІДЖЕННЯ РЕЖИМІВ ЗАСТОСУВАННЯ БЛОКОВИХ СИМЕТРИЧНИХ ШИФРІВ ВІДПОВІДНО ДО ISO/IEC 10116-2006

1. Вступ

Криптографічний захист інформації є важливою складовою у забезпеченні інформаційної безпеки держави, тому дослідження сучасних методів криптографічного перетворення та обґрунтування перспективних напрямків зі створення національних технологій на рівні світових аналогів є надзвичайно складним та важливим науково-технічним завданням.

Найбільший розвиток і застосування у системах захисту інформації знайшли симетричні криптографічні перетворення [1 – 5]. Втім рівень захищеності інформаційних ресурсів залежить не тільки від властивостей застосовуваного блокового симетричного шифру (БСШ), але і від способів його використання, тобто властивості криптографічного перетворення безпосередньо залежать від режиму застосування БСШ [6, 7].

В цій роботі досліджуються сучасні режими застосування симетричних криптоперетворень, які визначені в міжнародному стандарті ISO/IEC 10116-2006. Аналізуються особливості застосування БСШ у різних режимах та досліджуються їх криптографічні властивості, зокрема проводяться статистичні дослідження послідовностей псевдовипадкових бітів, які сформовані із використанням різних режимів застосування БСШ ГОСТ 28147-89, TDEA, FIPS-197, Camellia та Калина.

2. Режими застосування БСШ відповідно до ISO/IEC 10116-2006

Під режимом шифрування зазвичай розуміється такий метод застосування БСШ, який дозволяє реалізувати перетворення послідовність блоків відкритих даних в послідовність блоків зашифрованих даних із отриманням певних, наперед визначених криптографічних властивостей. Міжнародний стандарт ISO/IEC 10116-2006 «Information technology – Security techniques – Modes of operation for an n-bit block cipher» дає специфікацію п’яти режимів застосування БСШ, а саме [6]:режим простої заміни або режим електронної кодової книги (Electronic Codebook Mode – ECB);режим гамування або режим лічильника (Counter Mode – CTR);режим гамування зі зворотнім зв'язком по шифр тексту (Cipher Feedback mode – CFB); режим зчеплення шифроблоків (Cipher Block Chaining – CBC); режим гамування зі зворотнім зв’язком по шифр гамі (Output Feedback – OFB). Розглянемо ці режими біль докладніше.

2.1. Режим простої заміни (Electronic Codebook Mode – ECB). Загальна схема шифрування та розшифрування у режимі простої заміни зображена на рис. 1 [6, 7]. Це найпростіший з точки зору реалізації режим застосування блокового симетричного шифрування, який полягає у забезпеченні конфіденційності окремих блоків відкритого тексту шляхом їх шифрування за введеним секретним ключем *K*. Тобто кожному блоку відкритого тексту співставляється блок шифр тексту, що є деякою аналогією призначення кодового слова в шифрувальній книзі.

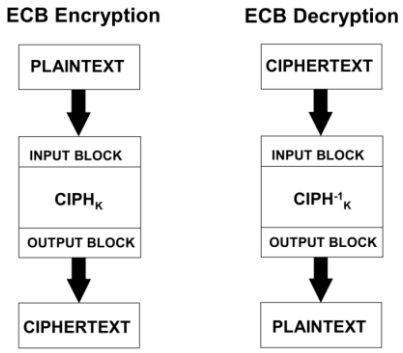

Рис. 1. Схема шифрування та розшифрування у режимі простої заміни

Режим ECB визначається наступним чином:

– шифрування

for $j = 1 \ldots n$

$C_j = CIPH_K(P_j)$;

– розшифрування

for $j = 1 \ldots n$

$P_j = CIPH^{-1}_K(C_i)$.

При застосуванні режиму ECB кожен виклик функції шифрування $CIPH_K(P_i)$ застосовується безпосередньо і незалежно до кожного блоку відкритого тексту P_i. Результуюча послідовність вихідних блоків C_j є зашифрованим текстом.

При розшифруванні у режимі простої заміни зворотна функція шифру $CIPH^{-1}_K(C_i)$ застосовується безпосередньо і незалежно до кожного блоку зашифрованого тексту P_j. Результуюча послідовність вихідних блоків P_i є відкритим текстом.

При застосуванні ECB багаторазові виклики функцій $CIPH_K(P_i)$ та $CIPH^{-1}_K(C_i)$ можуть бути реалізовані паралельно. Це є основною перевагою цього режиму. У цьому режимі для введеного секретного ключа будь-який блок відкритого тексту при шифруванні завжди перетворюється у один і той самий блок зашифрованого тексту. Якщо ця властивість не є бажаною в специфічному застосуванні, тоді режим простої заміни використовувати не потрібно. Режим простої заміни є обов'язковою складовою для всіх інших режимів застосування блокового симетричного шифрування.

2.2. Режим лічильника (Counter Mode – CTR). Режим CTR подібний до схеми шифрування у другому режимі застосування (гамування) алгоритму ГОСТ 28147-89 [1].

Режим CTR призначений для забезпечення конфіденційності шляхом швидкого шифрування блоків відкритого тексту із можливістю застосування паралельних обчислень. Він полягає в шифруванні набору вхідних блоків, які є виходом лічильника та у додаванні (за допомогою операції «ВИКЛЮЧНОГО АБО»⊕) результуючих блоків (блоків гами) до блоків відкритого тексту. Кожен блок на виході лічильника повинен бути відмінний від іншого блоку. Ця властивість не обмежується на одне інформаційне повідомлення, тобто для всіх повідомлень, які підлягають шифруванню із застосуванням одного і того самого ключа, блоки виходу лічильника повинні відрізнятися один від одного.

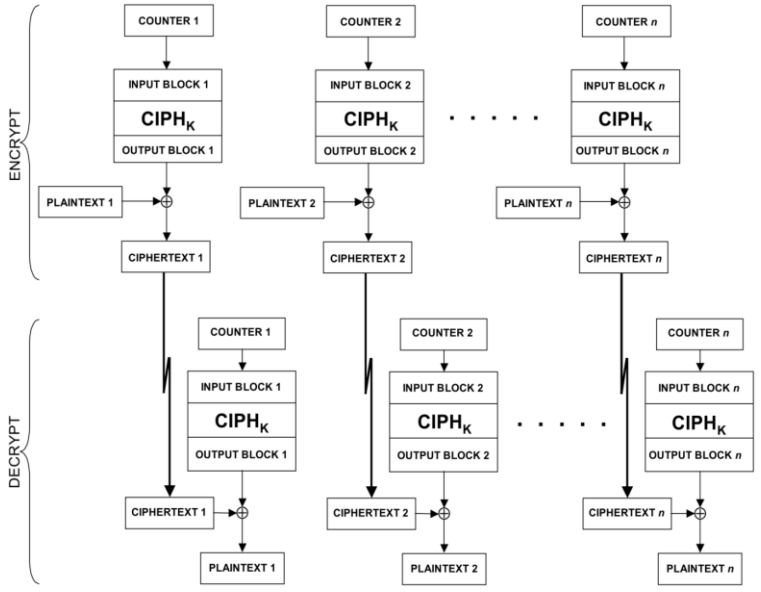

Рис. 2. Схема шифрування та розшифрування у режимі лічильника

Позначимо блоки з виходу лічильника як $T_1, T_2, \ldots, T_n$. Режим CTR визначається таким чином (рис. 2) [6, 7]:

– шифрування

for $j = 1, 2 \ldots n$

$O_j = CIPH_K(T_j)$;

for $j = 1, 2 \ldots n\text{-}1$

$C_j = P_j \oplus O_j$;

$C^*_n = P^*_n \oplus MSB_u(O_n)$;

– розшифрування

for $j = 1, 2 \ldots n$

$O_j = CIPH_K(T_j)$;

for $j = 1, 2 \ldots n\text{-}1$

$P_j = C_j \oplus O_i$;

$P^*_n = C^*_n \oplus MSB_u(O_n)$.

При шифруванні у режимі CTR кожен виклик функції шифрування $CIPH_K(T_j)$ застосовується безпосередньо і незалежно до кожного вихідного блоку лічильника T_j. Результуюча послідовність вихідних блоків O_j додається (через операцію⊕) до блоків відкритого тексту P_j, результат чого є послідовністю блоків зашифрованого тексту C_j. До останнього блоку відкритого тексту P_n, який, можливо, є частковим блоком, що заповнений на u бітів, додається (через операцію⊕) u найбільш значущих бітів (u найлівіших бітів) блоку O_n. Залишок з b-u бітів блоку O_n відкидається. Розшифрування у режимі CTR виконується аналогічно, тобто кожен виклик функції шифрування $CIPH_K(T_j)$ застосовується безпосередньо і незалежно до кожного вихідного блоку лічильника T_j. Після чого результуюча послідовність вихідних блоків O_j додається (через операцію⊕) до блоків шифр тексту C_j, результат чого є послідовністю блоків відкритого тексту P_j. До останнього блоку шифр тексту C_j, який, можливо, є частковим блоком заповненим

на u бітів додається (через операцію $\oplus$) u найбільш значущих бітів (u найлівіших бітів) блоку O_n. Залишок з b-u бітів блоку O_n відкидається.

2.3. Режим гамування зі зворотнім зв'язком по шифртексту (Cipher Feedback mode – CFB). Режим гамування зі зворотнім зв'язком по шифртексту подібний до схеми шифрування у третьому режимі алгоритму ГОСТ 28147-89, загальна схема CFB зображена на рис. 3 [6, 7].

Цей режим призначено для забезпечення конфіденційності шляхом шифрування потоку даних з розмноженням помилок і унеможливленням маніпуляцій із окремими блоками відкритого тексту. В основі CFB лежить застосування зворотного зв'язку, який реалізовано за шифртекстом, тобто до кожного блока відкритого тексту додається (через операцію $\oplus$) результат шифрування попереднього шифртексту. Фактично цей режим повторює CTR та відрізняється від нього правилом формування гами – вона формується не як результат шифрування вихідних послідовностей лічильника, а як результат шифрування попереднього блоку шифр тексту.

Режим CFB вимагає початкового значення (вектора ініціалізації) *IV* як початкове значення при формуванні гами. Цей вектор може не бути секретним, але він повинен бути непередбачуваним. Режим CFB також вимагає введення додаткового параметру – цілого числа s, такого, що $1 \le s \le b$. За специфікацією цього режиму кожен сегмент відкритого тексту $P^{\#}_j$ і кожен сегмент шифр тексту $C^{\#}_j$ складається з s бітів. Значення s іноді вказується в назві режиму, наприклад, 1-розрядний режим CFB, 8-розрядний режим CFB, 64-розрядний режим CFB, або 128-розрядний режим CFB.

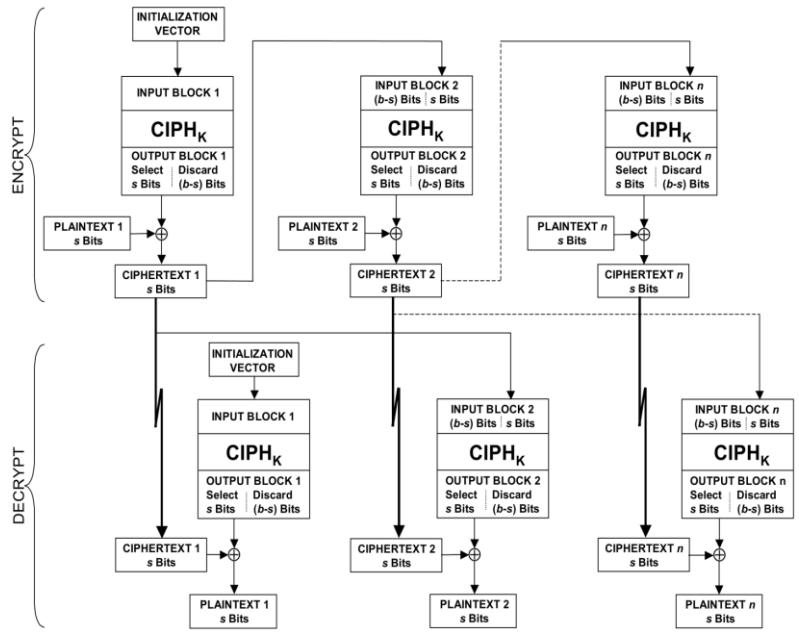

Рис. 3. Схема шифрування та розшифрування у режимі гамування зі зворотнім зв'язком по шифртексту

Формально режим CFB визначається наступним чином [6, 7]:

– шифрування

$I_1 = IV$;

for $j = 2 \dots n$

$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$;

for $j = 1, 2 \dots n$

$O_j = CIPH_K(I_j)$;

for $j = 1, 2 \dots n$

$C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j)$;

– розшифрування

$I_1 = IV$;

for $j = 2 \dots n$

$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$

for $j = 1, 2 \dots n$

$O_j = CIPH_K(I_j)$;

for $j = 1, 2 \dots n$

$P^{\#}_j = C^{\#}_j \oplus MSB_s(O_j)$.

Першим вхідним блоком для формування гами є вектор ініціалізації, тобто $I_1 = IV$.

Застосування над ним функції шифрування дає блок гами O_j, s старших біт якої додається (через операцію$\oplus$) до першого блоку відкритого тексту, тобто $C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j)$.$b$-$s$ молодших розрядів *IV* конкатенуються з s бітами першого сегменту за шифр тексту, щоб сформувати другий вхідний блок $I_j = LSB_{b-s}(I_{j-1}) | C^{\#}_{j-1}$ для генерації другого блоку гами $O_j = CIPH_K(I_j)$. Альтернативний спосіб утворення другого вхідного блоку I_j полягає у тому, що біти перших вхідних блоків циклічно зсуваються на s позицій вліво, а потім сегмент зашифрованого тексту замінює s молодших розрядів результату.

Описаний процес повторюється з послідовними вхідними блоками, поки не будуть сформовані всі сегменти зашифрованого тексту. Кожен вхідний блок I_j, який призначено для формування блоку гами, шифрується, а s старших біт у якості гами додаються (через опера-

цію⊕) до блоку відкритого тексту. Результатом є сегмент шифр тексту, який застосовується для формування наступного блоку гами. Таким чином, при формуванні гами реалізується зворотній зв'язок із попереднім сегментом шифр тексту. Цей зворотний зв'язок може бути описаний в термінах індивідуальних сегментів: якщо $I_j = (i_1, i_2, ..., i_b)$ є j-й вхідний блок для формування j-го блоку гами, і $C_j = (c_1, c_2, ..., c_s)$є j-й сегмент зашифрованого тексту, тоді $(j+1)$- вхідним блоком є$I_j = (i_{s+1}, i_{s+2}, ..., i_b, c_1, c_2, ..., c_s)$.

При розшифруванні по шифртексту першим вхідним блоком для формування гами є вектор ініціалізації, тобто $I_1 = IV$. Кожний наступний вхідний блок для формування гами, як і при шифруванні, зв'язується із b-s молодшими розрядами попереднього вхідного блоку $LSB_{b-s}(I_{j-1})$ та s старшими бітами попереднього зашифрованого сегменту $C^{\#}_{j-1}$, тобто $I_j = LSB_{b-s}(I_{j-1})| C^{\#}_{j-1}$. Функція шифрування застосовується до кожного вхідного блоку I_j, що дає блок гами $O_j = CIPH_K(I_j)$. s старших бітів гами додаються (через операцію ⊕) до відповідних сегментів шифр тексту $C^{\#}_j$, що дозволяє отримати сегмент відкритого тексту $P^{\#}_j = C^{\#}_j \oplus MSB_s(O_j)$.При шифруванні у режимі CFB вхідний блок I_j (крім першого $I_1 = IV$) залежить від результату шифрування попереднього блоку, тому шифрування окремих блоків не може бути реалізовано паралельно. При розшифруванні у якщо відомі відповідні блоки шифр тексту виклики функції шифруванні можуть бути реалізовані паралельно.

2.4. Режим зчеплення шифрблоків (Cipher Block Chaining – CBC). Режим зчеплення шифр блоків призначено для забезпечення конфіденційності, у якому процес шифрування побудовано шляхом об'єднання («формування ланцюжка») блоків відкритого тексту з попередніми блоками зашифрованого тексту.

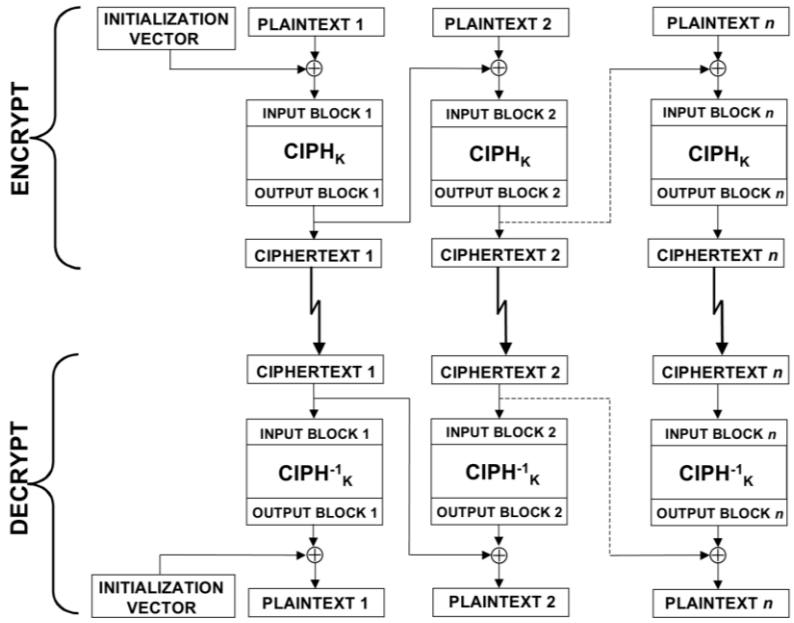

Рис. 4. Схема шифрування та розшифрування у режимі зчеплення шифрблоків

Режим CBC вимагає, щоб вектор ініціалізації IV був об'єднаний з першим блоком відкритого тексту. При цьому вектор IV не повинен бути секретним, але він повинен бути непередбаченим.

Режим CBC визначається наступним чином (рис. 4) [6, 7]:

– шифрування

$C_1 = CIPH_K(P_1 \oplus IV);$

for $j = 2 \ldots n$

$C_j = CIPH_K(P_j \oplus C_{j-1});$

– розшифрування

$P_1 = CIPH^{-1}_K(C_1) \oplus IV;$

for $j = 2 \ldots n$

$P_j = CIPH^{-1}_K(C_j) \oplus C_{j-1}.$

Перший вхідний блок шифрування формується як результат ВИКЛЮЧНОГО-АБО (тобто через операцію⊕) першого блоку відкритого тексту з вектором ініціалізації IV. Функція шифрування застосовується до першого вхідного блоку, її виходом є перший блок зашифрованого тексту. Цей вихідний блок також додається (через операцію⊕) до другого блока даних відкритого тексту щоб сформувати другий вхідний блок шифрування. Далі знов застосовується функція шифрування, яка формує другий вихідний блок, який є другим блоком зашифрованого тексту. Вихідний блок додається через виключне-АБО до блоку звичайного тексту щоб сформувати наступний вхідний блок і операції повторюються. Таким чином, кожен послідовний блок відкритого тексту додається через виключне-АБО до попереднього результату шифрування,

щоб сформувати новий вхідний блок. Функція шифрування застосовується до кожного вхідного блоку, щоб сформувати блок зашифрованого тексту.

При розшифруванні у режимі зчеплення шифр блоків, функція розшифрування застосовується до першого блоку зашифрованого тексту, її результатом є вихідний блок, який додається через операцію виключного-АБО (операцію⊕) з вектором ініціалізації IV, щоб сформувати перший блок відкритого тексту. Зворотна функція шифру також застосовується до другого блоку зашифрованого тексту, її вихідний блок через операції виключного-АБО додається до першого блоку зашифрованого тексту, щоб сформувати другий блок відкритого тексту. Взагалі, щоб сформувати будь-який блок (крім першого) відкритого тексту, зворотна функція шифру застосовується до відповідного блоку зашифрованого тексту, і результуючий блок додається через виключне-АБО з попереднім блоком зашифрованого тексту.

При шифруванні у режимі зчеплення шифр блоків кожний вхідний блок шифрування залежить від результату попередньої переднього шифрування, тому шифрування неможе виконуватися в паралелі. Проте при розшифруванні вхідні блоки (блоки зашифрованого тексту), доступні негайно, тому багаторазове розшифрування може виконуватися в паралелі.

2.5. Режим гамування зі зворотнім зв'язком по шифргамі (Output Feedback – OFB). Режим гамування зі зворотнім зв'язком по шифр гамі призначено для забезпечення конфіденційності. Цей режим засновано на шифруванні вектору ініціалізації IV для генерації послідовності вихідних блоків (шифргами), які додаються через операцію ВИКЛЮЧНОГО-АБО (операцію⊕)до звичайного тексту, щоб сформувати зашифрований текст і навпаки, до шифр тексту для його розшифрування. Вектор ініціалізації IV повинен бути унікальним для кожного застосування із наданим (фіксованим) ключем.

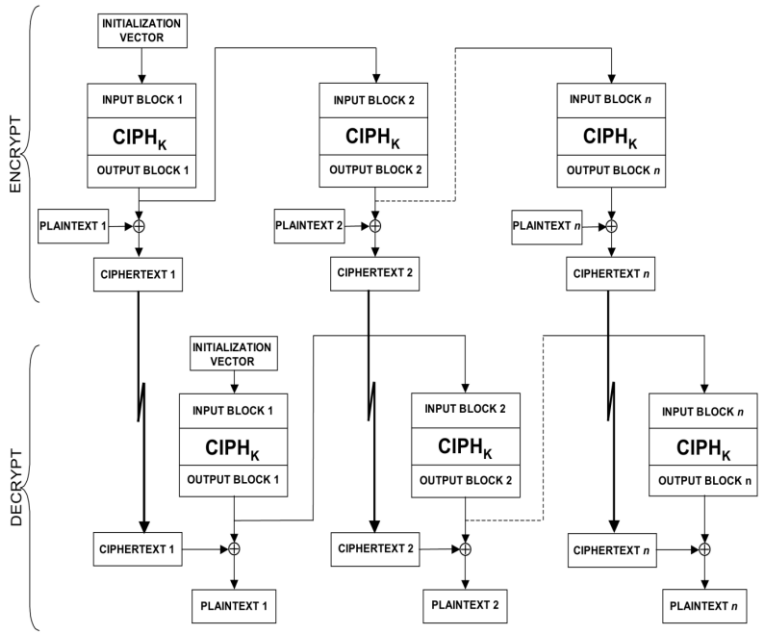

Рис. 5. Схема шифрування та розшифрування у режимі гамування зі зворотнім зв'язком по шифргамі

Режим OFB визначається наступним чином (рис. 5) [6, 7]:

– шифрування

$I_1 = IV;$

for $j = 2 \dots n$

$I_j = O_{j-1};$

for $j = 1, 2 \dots n$

$O_j = CIPH_K(I_j);$

for $j = 1, 2 \dots n\text{-}1$

$C_j = P_j \oplus O_j;$

$C^*_n = P^*_n \oplus MSB_u(O_n);$

– розшифрування

$I_1 = IV;$

for $j = 2 \dots n$

$I_j = O_{j-1};$

for $j = 1, 2 \dots n$

$O_j = CIPH_K(I_j);$

for $j = 1, 2 \dots n\text{-}1$

$P_j = C_j \oplus O_j;$

$P^*_n = C^*_n \oplus MSB_u(O_n).$

При шифруванні до вектору ініціалізації IV застосовується функція шифрування для отримання першого вихідного блоку (першого блоку гами). Перший блок гами через операцію ВИКЛЮЧНОГО-АБО додається до першого блоку відкритого тексту, результатом є перший блок зашифрованого тексту. Функція шифрування застосовується знов на першому блоці гами для отримання другого блоку гами. Отриманий другий блок гами через операцію ВИКЛЮЧНОГО-АБО додається до другого блоку відкритого тексту, як результат – другий блок зашифрованого тексту. Функція шифрування викликається знову і операції повторюються. Таким чином, послідовні вихідні блоки (блоки гами) генеруються шляхом застосування функції шифрування до попередніх блоків гами, які через операцію виключного-АБО дода-

ються до відповідних блоків відкритого тексту для формування блоків зашифрованого тексту. До останнього блоку відкритого тексту, який, можливо, є частково заповненим блоком із u бітів, додається через операцію ВИКЛЮЧНОГО-АБО u найбільш значущих бітів останнього блоку гами; b-u бітів останнього блоку гами, що залишилися, відкидаються.

При розшифруванні до вектору ініціалізації IV застосовується функція шифрування для формування першого блоку гами (першого вихідного блоку). Перший блок гами через операцію ВИКЛЮЧНОГО-АБО додається до першого блоку зашифрованого тексту, результатом є перший блок відкритого тексту. Перший блок гами обробляється функцією шифрування для формування другого блоку гами. Другий блок гами через операцію ВИКЛЮЧНОГО-АБО додається до другого блоку зашифрованого тексту для отримання другого блоку відкритого тексту. Другий блок гами згову обробляється функцією шифрування і операція повторюється. Таким чином, послідовні вихідні блоки (блоки гами) генеруються шляхом застосування функції шифрування до попередніх блоків гами, які через операцію ВИКЛЮЧНОГО-АБО додаються до відповідних блоків зашифрованого тексту для формування блоків відкритого тексту. До останнього блоку зашифрованого тексту, який, можливо, є частково заповненим блоком із u бітів, додається через операцію виключного-АБО u найбільш значущих бітів останнього блоку гами; b-u бітів останнього блоку гами, що залишилися, відкидаються.

Як при шифруванні, так і при розшифруванні у режимі гамування зі зворотнім зв'язком по шифргамі кожен результат (крім першого) функції шифрування залежить від результатів попередньої переднього шифрування. Тому багаторазові виклики функції шифрування не можуть виконуватися паралельно. Проте, якщо вектор ініціалізації IV відомий, блоки гами можуть генеруватися заздалегідь до отримання даних відкритого тексту або зашифрованого тексту. Режим OFB вимагає унікальне значення вектору ініціалізації IV для кожного повідомлення, яке буде зашифроване коли-небудь із наданим ключем. Якщо цю вимогу буде порушено, тобто вектор ініціалізації IV буде використано для шифрування більш ніж одного повідомлення, конфіденційність таких повідомлень, можливо, буде поставлено під загрозу. Зокрема, якщо відомо блок звичайного тексту будь-якого з цих повідомлень, скажімо, *j*-й блок звичайного тексту, тоді *j*-й блок гами може бути легко визначений від *j*-го блоку зашифрованого тексту повідомлення. Ця інформація дозволяє отримати *j*-й блок відкритого тексту з *j*-го блоку шифр тексту для будь-якого іншого повідомлення, яке зашифроване із використанням такого ж самого вектору ініціалізації IV. Конфіденційність, можливо, так само буде поставлена під загрозу, якщо будь-який з блоків гами використовується у якості вектору ініціалізації при шифруванні іншого повідомлення з наданим ключем.

3. Результати статистичних досліджень різних режимів застосування БСШ ГОСТ 28147-89, TDEA, FIPS-197, Camellia та Калина.

Проведемо статистичні дослідження послідовностей псевдовипадкових бітів, які сформовані із використанням різних режимів застосування БСШ, зокрема оцінимо статистичну безпеку шифрів ГОСТ 28147-89, TDEA, FIPS-197, Camellia та Калина у режимах шифрування, визначених у ISO/IEC 10116-2006. При проведенні статистичних досліджень будемо використовувати запропонований у [8] підхід, щодо оцінювання математичного очікування числа пройдених статистичних тестів за методикоюNISTSTS [9].

Методика тестування складається зі 100 проходів створення вихідної послідовності генератора випадкових біт та подальшого тестування цих послідовностей. Для створення вихідної послідовності у генераторі використовуються набор з 100 ключів, створених апаратним генератором випадкових бітів. Цей набор є для всіх однаковим. Вхідна послідовність представляю собою вектор даних, який складається з 64 бітних блоків, згенерованих модифікованим алгоритмом лічильника, що описаний далі:

- вхідний 64 блок послідовності поділяється на байти по 8 *бітів*. В кожному байті може бути лише 1 *біт* «1»;

- початковий стан блока представляє собою значення «1», записане у вигляді 64-бітного блоку;

- кожний *i*-й блок (для i >1) є циклічним зсувом в байтах з переносом (i – 1) блока (рис. 6);

i-1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***
i	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***1***	***0***

Рис. 6. Приклад формування *i*-го блоку за відомим (i – 1) блоком

- на кожному кроці відбувається циклічний зсув «1» біта в молодшому байті. Коли біт «1» доходить до кінця свого байту, то відбувається перенос цього біта в наступний байт та циклічний зсув у цьому байті. Якщо наступний байт є нульовим, то «1» біт встановлюється як молодший (рис. 7);

i-1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***	***0***	***0***	***0***	***0***	***0***	***0***	***0***
i	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***

Рис. 7. Приклад формування *i*-го блоку за відомим (i – 1) блоком з переносом біта в наступний байт

- якщо байт не є нульовим, то відбувається циклічний зсув за тими ж правилами, що і в молодшому байті (рис. 8).

i-1	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***	***1***	***0***	***0***	***0***	***0***	***0***	***0***	***0***
i	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	...	*0*	*0*	***0***	***0***	***0***	***0***	***0***	***0***	***1***	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***0***	***1***

Рис. 8. Приклад формування *i*-го блоку за відомим (i – 1) блоком без переносу біта в наступний байт

Алгоритмічно цей процес використовує наступну або еквівалентну послідовність кроків:

1) block_0 = 1;
2) для i = 1 .. n:

a. block_i = block_i-1<<< 1;

де операція <<< є циклічний зсув з переносом:

1) j = 0;
2) Якщо block [j] ≠ 0x80: (перевірка на положення «1» біту)

Перехід до кроку 5)

В іншому випадку:

Перехід до кроку 3)

3) block [j] = 1;

j = j + 1;

перехід на крок 4).

4) Якщо block [j] ≠ 0; (перевірка, чи є у *j*-му байті блоку «1» біт)

Перехід до кроку 2)

В іншому випадку

block [j] = 1;

Повернуті значення block

5) block [j] << 1;

Повернути значення block

де: block_i – *i*-й блок послідовності, block [j] – *j*-й байт в блоку (вважається, що block [0] – наймолодший байт в блоку, а block [7] – найстарший байт в блоку).

Перевірка на вихід за межі block не виконується, бо цей алгоритм дає можливість згенерувати послідовність більш ніж $1{,}2 * 10^9$ *біт*.

Отримані результати експериментальних досліджень зведено в табл.1.

Таблиця 1

Результати експериментальних досліджень при N = 100

	M099	D099	S099	P099	M096	D096	S096	P096	Min096
ГОСТ_ECB	132.34	57.50	7.58	0.99	186.52	1.65	1.28	1.00	182
ГОСТ_CBC	130.77	65.76	8.11	0.99	186.56	2.01	1.42	1.00	181
ГОСТ_CFB	133.49	61.77	7.86	0.99	186.70	1.99	1.41	1.00	178
ГОСТ_CTR	131.86	69.64	8.35	0.99	186.99	1.57	1.25	1.00	183
ГОСТ_OFB	132.78	70.21	8.38	0.98	186.70	2.59	1.61	1.00	180
AES_ECB	133.96	51.34	7.17	0.99	186.89	1.24	1.11	1.00	183
AES_CBC	132.62	40.38	6.35	1.00	186.66	1.88	1.37	1.00	181
AES_CFB	133.02	57.80	7.60	0.99	186.76	2.02	1.42	1.00	182
AES_CTR	133.59	49.52	7.04	1.00	186.55	2.19	1.48	1.00	182
AES_OFB	133.07	53.49	7.31	0.99	186.75	1.37	1.17	1.00	183
KALINA_ECB	132.16	56.73	7.53	0.99	186.82	1.57	1.25	1.00	183
KALINA_CBC	131.53	54.81	7.40	0.99	186.73	1.74	1.32	1.00	182
KALINA_CFB	133.60	54.60	7.39	0.99	186.96	1.28	1.13	1.00	184
KALINA_CTR	132.81	48.29	6.95	1.00	186.83	2.06	1.44	1.00	182
KALINA_OFB	132.39	50.64	7.12	0.99	186.78	1.75	1.32	1.00	181
CAMELLIA_ECB	132.76	59.80	7.73	0.99	186.76	1.78	1.34	1.00	182
CAMELLIA_CBC	131.79	58.21	7.63	0.99	186.90	1.83	1.35	1.00	182
CAMELLIA_CFB	133.82	58.37	7.64	0.99	186.71	1.37	1.17	1.00	183
CAMELLIA_CTR	131.86	50.74	7.12	0.99	186.78	1.33	1.15	1.00	181
CAMELLIA_OFB	132.96	49.26	7.02	1.00	186.96	1.92	1.39	1.00	181
TDES_ECB	133.28	66.52	8.16	0.99	186.60	1.98	1.41	1.00	180
TDES_CBC	132.67	54.50	7.38	0.99	187.01	1.49	1.22	1.00	182
TDES_CFB	131.89	51.96	7.21	0.99	186.68	1.54	1.24	1.00	183
TDES_CTR	132.64	63.23	7.95	0.99	186.87	1.97	1.40	1.00	181
TDES_OFB	131.54	63.57	7.97	0.99	186.86	1.74	1.32	1.00	180

У табл. 1 використано такі позначення:

– ГОСТ_ECB, ГОСТ_CBC, ГОСТ_CFB, ГОСТ_CTR, ГОСТ_OFB – реалізація БСШ ГОСТ 28147-89 в режимах ECB, CBC, CFB, CTR та OFB відповідно;

– AES_ECB, AES_CBC, AES_CFB, AES_CTR, AES_OFB – реалізація БСШ FIPS-197 (AES) в режимах ECB, CBC, CFB, CTR та OFB відповідно;

– KALINA_ECB, KALINA_CBC, KALINA_CFB, KALINA_CTR, KALINA_OFB – реалізація БСШ Калина в режимах ECB, CBC, CFB, CTR та OFB відповідно;

– CAMELLIA_ECB, CAMELLIA_CBC, CAMELLIA_CFB, CAMELLIA_CTR, CAMELLIA_OFB – реалізація БСШ Camellia в режимах ECB, CBC, CFB, CTR та OFB відповідно;

– TDES_ECB, TDES_CBC, TDES_CFB, TDES_CTR, TDES_OFB – реалізація БСШ TDEA в режимах ECB, CBC, CFB, CTR та OFB відповідно;

У табл. 1 наведені такі дані (методику досліджень докладно викладено в [8]):

– M096 M099 – оцінки математичного сподівання (вибіркові середні) числа пройдених статистичних тестів за критерієм $P_j \geq 0{,}96$ і за критерієм $P_j \geq 0{,}99$ відповідно;

– D096 D099 і (S096 і S099) – оцінки дисперсій (середньоквадратичних відхилень) результатів тестування числа пройдених статистичних тестів за критеріями $P_j \geq 0{,}96$ і $P_j \geq 0{,}99$ відповідно;

– P099 – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0{,}99$ і точності $\varepsilon = 2$;

– P096 – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0{,}96$ і точності $\varepsilon = 1$.

В останній колонці "Min096" табл. 1 наведені мінімальні значення числа пройдених статистичних тестів за критерієм $P_j \geq 0{,}96$.

При проведенні експериментальних досліджень використані наступні параметри БСШ:

– довжина вхідних блоків БСШ Калина і Камелія дорівнює 128 *бітів*;

– довжина ключа БСШ FIPS-197, Камелія і Калина дорівнює 256 *бітів*.

Аналіз отриманих результатів свідчить, що досліджувані шифри мають дуже високі показники статистичної безпеки. Для всіх випадків отримані результати з високим проходженням статистичних тестів як за критерієм $P_j \geq 0{,}96$ і за критерієм $P_j \geq 0{,}99$. У всіх досліджуваних режимах застосування БСШ (ECB, CBC, CFB, CTR та OFB) формовані вихідні послідовності за критерієм $P_j \geq 0{,}99$ проходять від 130 до 134 статистичних тестів, та за критерієм $P_j \geq 0{,}96$ проходять від 185 до 187 статистичних тестів. Оцінки математичних сподівань (вибіркових середніх) числа пройдених статистичних тестів отримано з високою точністю ($\varepsilon = 2$ та $\varepsilon = 1$, відповідно) та вірогідністю.

Висновки

Отримані результати експериментальних досліджень підтверджують загальний висновок щодо високих показниках статистичної безпеки відомих шифрів ГОСТ 28147-89, TDEA, FIPS-197, Camellia та Калина. У всіх досліджених режимах шифрування, які визначено у міжнародному стандарті ISO/IEC 10116-2006, розглянуті БСШ показали високе число пройдених статистичних тестів за методикою NISTSTS. Запропонована у роботі [8] методика статистичного тестування криптографічних алгоритмів дозволяє проводити таку оцінку із високою точністю та достовірністю.

Перспективним в подальшому є аналіз та експериментальні дослідження режимів шифрування, які визначено в стандарті США NISTSpecialPublication 800-38:"BlockCipherModes" [7], обґрунтування режимів застосування БСШ, які можуть бути запропоновані у якості національного стандарту України.

Список літератури: 1. *ГОСТ 28147-89.* Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М. : Изд-во стандартов, 1989. – 20 с. 2. *National* Institute of Standards and Technology, “NIST Special Publication 800-67: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher”, January 2012. Режим доступу: http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf. 3. *National* Institute of Standards and Technology, “FIPS-197: Advanced Encryption Standard”, November 2001. Режим доступу:

http://csrc.nist.gov/publications/ fips/fips197/fips-197.pdf/ 4. *M. Matsui et al*. A Description of the Camellia Encryption Algorithm, Internet Engineering Task Force, Request for Comment 3713, April 2004. Режим доступу: http://www.ietf.org/rfc/rfc3713.txt 5. *Горбенко, І. Д.* Перспективний блоковий симетричний шифр «Калина»: основні положення та специфікації / І. Д. Горбенко, В. І. Долгов, Р. В. Олейніков [та ін.] // Прикладна радіоелектроніка. – 2007. – Т. 6, № 2. – С. 195–208. 6. *ISO/IEC 10116*. Information technology – Security techniques – Modes of operation for an n-bit block cipher. [Електронний ресурс]. Режим доступу: http://www.iso.org .7. *NIST Special* Publication 800-38. Block Cipher Modes. Режим доступу:http://csrc.nist.gov/groups/ST/toolkit/BCM/current_modes.html. 8. *Кузнецов, А.А.* Методика статистического тестирования криптографических алгоритмов / Кузнецов А.А., Мордвинов Р.И., Колованова Е.П., Самойлова А.В. // Радиотехника. – 2014. – Вып.176. – С.40 – 44. 9. *Special* Publication 800 22. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Режим доступу: http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf.

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 23.12.2013