транспьютерной системе // Транспьютерные системы и их применения Тез. докл. 2-й конф российской транспьютерной ассоциации, Домодедово, 1992.- С. 32. 3. Жуков И. А., Юрьев Ю. Н., Балашов А. Ю. Вычислительная структура для решения систем линейных алгебраических уравнений // Микропроцессорные системы и персональные ЭВМ.- Киев: Ин-т кибернетики АН Украины 1994.- С. 41-44.

УДК 681.3

СТРУКТУРНА ОРГАНІЗАЦІЯ АЛГОРИТМІВ СИМЕТРИЧНОГО БЛОКОВОГО ШИФРУВАННЯ

Тимур Коркішко
Тернопільська академія народного господарства

Анотація: **Досліджено базові підходи та особливості структурної організації сучасних алгоритмів симетричного блокового шифрування (АСБШ). Розроблено та досліджено структурну організацію узагальненого АСБШ. Запропоновано та досліджено структурну організацію складових процедур АСБШ, їх класифікацію, аналітичні вирази та графічну інтерпретацію. Наведені та проаналізовані основні підходи до виконання складових процедур АСБШ.**

Summary: **Base approaches and peculiarities of the structural organization of the modern symmetric block encryption algorithm (SBEA) are investigated. A structural organization of generalized SBEA is developed and analyzed. The structural organization of the components of the SBEA procedures, their classification, the analytical equations and the graphical interpretation are proposed and investigated. The main approaches of the SBEA component parts execution are considered and analyzed.**

Ключові слова: **Алгоритм симетричного блокового шифрування, структурна організація алгоритму шифрування.**

I Вступ

Використання алгоритмів симетричного блокового шифрування (АСБШ) в областях, де необхідно здійснювати передавання, збереження та накопичення даних, дозволяє вирішити питання конфіденційності інформації. Для реалізації АСБШ використовуються програмні та апаратні комп'ютерні засоби. Відомо [1], що використання програмованих універсальних процесорів не забезпечує виконання процедур шифрування над інтенсивними потоками даних у реальному масштабі часу, тому обробка даних АСБШ здійснюється спеціалізованими процесорами, структура операційних пристроїв яких орієнтована на виконуваний алгоритм. Для створення ефективних спеціалізованих процесорів, орієнтованих на обробку даних у реальному масштабі часу, необхідно проведення аналізу структурної організації АСБШ та шляхів організації їх обчислень [2]. Попередні дослідження структурної організації АСБШ, наприклад в [3, 4, 5], проводилися на конкретних алгоритмах. Розв'язок актуальних задач розвитку архітектур спеціалізованих процесорів АСБШ та розробка принципів побудови їх операційних пристроїв вимагають досліджень базових підходів до побудови АСБШ, їх структурної організації, характеру обробки даних тощо. Тому дана робота присвячена дослідженню структурної організації узагальненого АСБШ, виділенню спільних компонент АСБШ та проведенню їх класифікації, дослідженню варіантів організації взаємодії складових компонент.

II Базові підходи до побудови алгоритмів симетричного блокового шифрування

Блоковим шифром називають функцію, яка відображає n-бітові вхідні блоки відкритого тексту у n-бітові блоки зашифрованого тексту, n називають довжиною блоку шифру [6]. Блоковий шифр можна розглядати як деяку параметризовану функцію підстановки елементів великого розміру. Параметром функції виступає k-бітовий ключ K, який належить простору ключів. Блоковий шифр є симетричним, якщо ключ зашифрування рівний ключу розшифрування. АСБШ реалізує функцію блокового шифру для зашифрування та розшифрування вхідних блоків.

Для розробки обчислювально стійких АСБШ використовують два загальних принципи [7]: розсіювання та перемішування. Розсіюванням називається розповсюдження впливу одного знаку відкритого тексту на велику кількість зашифрованого тексту, що зумовлює маскування статистичних властивостей початкового повідомлення. Перемішуванням називається шифруюче перетворення, яке порушує взаємозв'язки статистичних характеристик вхідного та вихідного текстів. Також використовується принцип розповсюдження впливу одного знаку ключа на велику кількість знаків зашифрованого тексту.

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

Складені АСБШ

Одним із способів досягнення доброго розсіювання та перемішування є побудова складеного шифру, який складається із ряду послідовно використовуваних простих шифрів, кожен з яких вносить невеликий вклад у перемішування чи розсіювання [8]. В таких складених шифрах процедури шифрування одного типу чергуються із процедурами другого типу. Простими шифрами можна обрати, наприклад, підстановки (*S*), перестановки (*T*) та лінійні перетворення (*L*) [7]. У цьому випадку результуючий шифр представляються у вигляді:

$$F = T_{Nr} L_{Nr} S_{Nr} \mathrm{K}\ T_2 L_2 S_2 T_1 L_1 S_1, \tag{1}$$

де *Nr* – кількість шарів простих шифрів.

Таємний ключ може бути використаний при виконанні процедур як одного певного типу (*T*, *L* чи *S*), так і декількох, чи усіх типів. Один із варіантів простого складеного шифру, що містить прості шифри підстановки та перестановки, приведений на рис. 1.

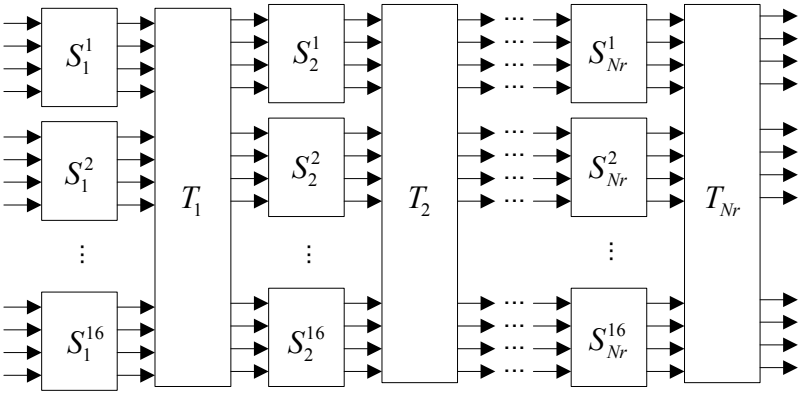

Рисунок 1 – Варіант складеного шифру на основі підстановок та перестановок

Тут блоки *S* позначають операції підстановки над 4-бітовими підблоками вхідного повідомлення, *T* – операцію перестановки над 64-бітовим блоком даних.

Процедура обробки даних даного шифру полягає у виконанні послідовності операцій підстановки та перестановки. Блок *S* реалізує заміну перетворюваного значення на нове. Заміна задається таблицею підстановки

$$\begin{pmatrix} 0 & 1 & 2 & \mathrm{K} & 15 \\ \alpha_0 & \alpha_1 & \alpha_2 & \mathrm{K} & \alpha_{15} \end{pmatrix}, \tag{2}$$

де стовпці задають відповідність між вхідним 4-бітовим значенням (верхній рядок) та вихідним 4-бітовим значенням (нижній рядок). Операції підстановки та перестановки є зворотними. Процедура розшифрування полягає у виконанні зворотних операцій у зворотному порядку.

У даному прикладі припускалося, що блоки *S* і *T* є залежними від ключа шифрування, тобто вони є таємними. З іншого боку можна запропонувати варіант цієї схеми, де блоки перестановок є відомими (фіксованими), а ключем є множина підстановок, які виконуються блоками *S*.

Для забезпечення більшої стійкості шифру проти атак різного виду [9, 10] доцільно виконувати підстановки над підблоками більшого розміру. Проте, у цьому випадку збільшується складність процедури вибору оптимальних підстановок та складність реалізації шифрів [11].

Ітеративні АСБШ

Велика кількість сучасних АСБШ, наприклад [12 – 35], використовує багатократне повторення деякого набору операцій перетворень, що називаються раундом шифрування [36]. В кожному раунді використовується деяка частина ключа шифрування, що називається набором підключів. Елемент набору підключів називається підключем. Правило формування наборів підключів із ключа шифрування називається розписом формування ключів та задається у вигляді деякого алгоритму. Ітеративний АСБШ можна описати рекурсивною формулою [11]:

$$B_i = E\left(B_{i-1}, \{Sk\}_i\right), \tag{3}$$

де E – раундова функція зашифрування, B_i, B_{i-1} – вихідний та вхідний блок для i-го раунду, $\{Sk\}_i$ – набір підключів, що використовуються у i-му раунді, i = 1, 2, …, Nr, Nr – кількість раундів перетворення зашифрування. У таких ітеративних алгоритмах раундова функція повинна мати обернення. Нехай D – раундова функція розшифрування. Тоді процедура розшифрування описується формулою:

$$B_i = D\left(B_{i-1}, \{Sk\}_{R-i+1}\right). \tag{4}$$

Для розробки блокових шифрів широко використовується криптографічна схема, запропонована Файстелем у [36] та розвинута у [37], яка дозволяє використовувати довільні (в тому числі і необоротні) функції F для побудови оборотних шифруючих перетворень. Суть цієї криптографічної системи полягає в наступному. Вхідний блок B розбивається на два однакових підблоки L (лівий підблок) та R (правий підблок). Зашифрування виконується у відповідності з такими рекурентними співвідношеннями:

$$\begin{aligned} L_i &= R_{i-1}, \\ R_i &= L_{i-1} \oplus F\left(R_{i-1}, \{Sk\}_i\right) \end{aligned} \tag{5}$$

де $L_0 \| R_0$ – вхідний блок даних, $L_{Nr} \| R_{Nr}$ – перетворений блок даних на виході останнього раунду зашифрування, $\oplus$ – операція порозрядного сумування по модулю два, | – операція конкатенації блоків, i = 1, 2, …, Nr. Схема одного раунду зашифрування зображена на рис. 2а.

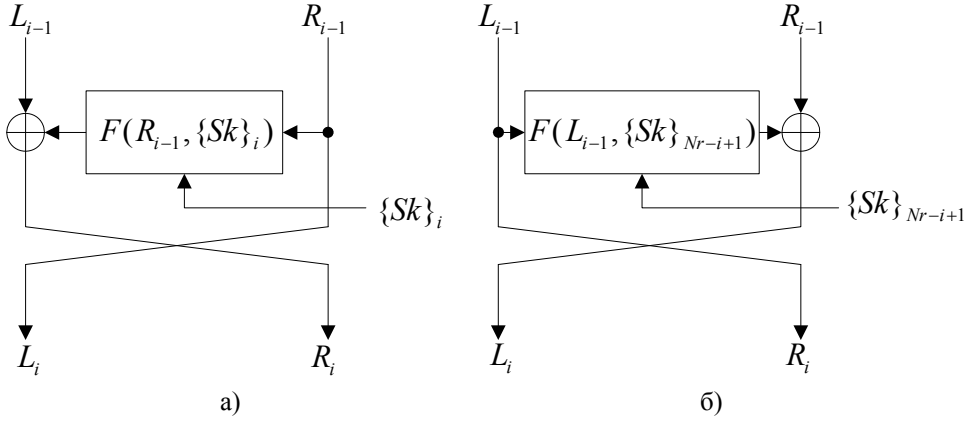

Рисунок 2 – Схема i-го раунду ітеративного шифру:
а) зашифрування, б) розшифрування

Розшифрування полягає у виконанні наступних Nr раундів:

$$\begin{aligned} R_i &= L_{i-1}, \\ L_i &= R_{i-1} \oplus F\left(L_{i-1}, \{Sk\}_{R-i+1}\right) \end{aligned} \tag{6}$$

Схема одного раунду розшифрування наведена на рис. 2б. Схема Файстеля не висуває вимоги існування оберненої до F функції, тому як функцію F можна використати довільні детерміновані процедури.

У деяких шифрах, наприклад [38], функція F задається табличним способом і процедура обчислення значення F виконується аналогічно до процедури обчислення значення блоку S. По аналогії із операціями підстановки таблиці для функції F деколи називаються блоками підстановки. В той же час операції підстановки типу F та типу S відрізняються наступним: розмір вхідного m та вихідного n блоків даних для підстановки типу F відрізняються, а для підстановки S вони рівні; розміром підстановки називаються значення m і n, підстановки S задають взаємно-однозначне відображення, підстановки F в загальному випадку можуть задавати відображення декількох вхідних різних значень в одне й те ж вихідних значень.

Розглянуті класичні підходи до побудови симетричних блокових шифрів передбачають використання деякої фіксованої множини операцій, на базі якої і будуються раундові функції. Ключ використовується для управління процесом шифрування шляхом визначення параметрів, що використовуються у шифруючих перетвореннях. Ціми параметрами служать вибрані за певним законом елементи ключа чи знаки псевдовипадкової послідовності, що генерується на основі ключа. Однак рядом дослідників були запропоновані альтернативні шляхи побудови процедур перетворення даних.

АСБШ із керованими операціями

Альтернативний підхід до побудови шифруючих перетворень передбачає використання керованих операцій [39]. При цьому код керування може формуватися як на основі перетворюваних даних, так і на основі підключа. Шифри такого виду належать до недетермінованих шифрів відносно використовуваних елементарних операцій (підстановка, перестановка, циклічний зсув тощо), конкретна модифікація яких на біжучому кроці обирається залежно від одного із підблоків даних, який піддається перетворенню. Ефективність використання керованих операцій збільшується із збільшенням числа потенційно можливих модифікацій, оскільки в цьому випадку розширюється підблок даних, над якими виконується результуюча операція [40, 41].

Використання керованої операції перестановки для перетворення підблоків передбачає, що один із підблоків використовується як керуючий. При цьому задається такий спосіб задання перестановки, при якому усім можливим значенням управляючого підблоку відповідають різні варіанти операції перестановки. Оскільки число можливих перестановок із n елементів складає $n! >> 2^n$, то існує спосіб, який задає унікальні перестановки для кожного значення управляючого n-бітового коду.

Як операцію перетворення, що залежить від даних, які перетворюються, можна використати операцію підстановки [11]. Тип підстановок може бути заданий як множина різних таблиць підстановок, кожній з яких присвоєний порядковий номер. Для виконання операції підстановки над одним підблоком використовують таблицю, номер якої вибирається залежно від значення деякого другого підблоку перетворюваних даних чи деякого двійкового вектора V, який формується залежно від перетворюваних даних. Наприклад, при використанні z таблиць підстановок номер таблиці підстановки можна обчислити за формулою $v = V \bmod 2^z$. Суть вибору таблиці підстановки, яка буде використовуватися на біжучому кроці, по спеціально сформованому двійковому вектору полягає у тому, щоб зробити вибір таблиці підстановки непередбаченим для кожного кроку перетворення підблоків, що підвищує стійкість криптографічного перетворення.

Нехай операції підстановки виконуються над підблоками цифрових даних розміром k біт. Тоді для визначення операції підстановки розміром $k \times k$ (вхідним вектором для операції є блок даних розміром k біт, вихідний блок також має розмір k біт) необхідно використання таблиці, яка містить два рядки чисел:

$$\begin{pmatrix} 0 & 1 & 2 & \text{K} & N-1 \\ \alpha_0 & \alpha_1 & \alpha_2 & \text{K} & \alpha_{N-1} \end{pmatrix},$$

де $N = 2^k$.

В даній таблиці у нижньому рядку присутні усі можливі значення k-бітового блоку рівно по одному разу, але у довільному порядку. Черговість розташування чисел у нижньому рядку визначає конкретний варіант таблиці підстановки, а тому і конкретний варіант операції підстановки, що виконується за допомогою цієї таблиці. Виконання операції підстановки виконується так: у верхньому рядку вибирається число, рівне значенню вхідного блоку; те число, яке знаходиться під цим значенням у нижньому рядку, береться за вихідний блок.

Номер використаної таблиці підстановки залежить від перетворюваних блоків і є невизначеним для біжучого кроку перетворення, тобто операція підстановки є невідома наперед для усіх раундів перетворення. Визначення конкретної операції підстановки проходить із участю таємного ключа та перетворюваного блоку.

У [42] була запропонована схема шифрування із механізмом виконання операцій, які залежать від оброблюваних даних та виконуються над підключами. Суть виконання операцій над підключами полягає у тому, що здійснюється модифікація підключів при шифруванні кожного вхідного блоку. У цьому випадку різні блоки відкритого тексту будуть шифруватися із використанням різних наборів модифікованих значень підключів. Така схема зумовлює модифікацію підблоків вхідних даних підключами, модифікованими за допомогою операції керованої перестановки. За управляючий код для операції перестановки використовуються другий підблок. Окрім перестановок загального виду для модифікації підключів можна використати циклічні зсуви, які, однак, є частковим випадком операції перестановки.

Незважаючи на альтернативні набори операцій перетворення даних, в загальному такий підхід до побудови АСБШ повторює складені та ітераційні структури обробки даних. Тому без втрати загальності будемо розглядати АСБШ класичного та альтернативного підходів в рамках одного узагальненого АСБШ. Його структурна організація, особливості складових процедур та організація взаємодії процедур при обробці даних наведені у наступному розділі.

III Структурна організація узагальненого алгоритму симетричного блокового шифрування

Аналіз структурної організації алгоритмів блокового шифрування [12 – 35, 43 – 46, 48] та базових способів їх побудови показав, що їх структурну організацію на високому рівні можна представити у вигляді поєднання двох основних компонент (рис. 3): процедури обробки даних та процедури обчислення розпису ключа. Порядок обробки блоку даних та ключа залежить від структури алгоритму та від коду виконуваної операції – шифрування чи розшифрування.

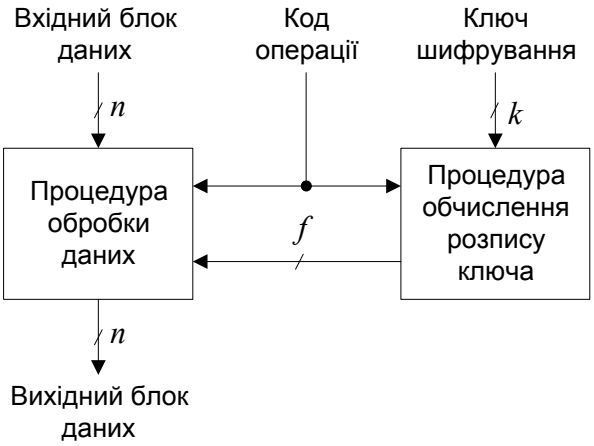

Рисунок 3 – Високорівневе представлення АСБШ

Процедура обробки даних призначена для перетворення вхідного блоку даних розміром n біт у вихідний блок даних згідно з описом АСБШ та складається із набору функціональних операторів перетворення даних. У загальному випадку структурна організація процедури обробки даних залежить від виконуваної операції, тобто процеси шифрування та розшифрування блоку даних різняться між собою, а алгоритми обчислення функцій шифрування та розшифрування не співпадають. При виконанні операцій шифрування процедура обробки даних використовує f біт ключової інформації – множину раундових підключів та дані для початкової та кінцевої модифікації вхідного та вихідного блоків, які формуються із ключа шифрування розміром k біт та деякої додаткової інформації, наприклад спеціально обраних констант, процедурою обчислення розпису ключа.

Аналогічно до процедури обробки даних, процедура обчислення розпису ключа складається із набору функціональних операторів перетворення вхідного ключа, а її структура залежить від виконуваної операції.

Виходячи з результатів аналізу структурних особливостей складових процедур розглянутих АСБШ та їх узагальнення, подамо ці складові як сукупність чотирьох процедур: процедури обробки даних для операції шифрування, процедури обробки даних для операції розшифрування, процедури обчислення розпису ключа для шифрування, процедури обчислення розпису ключа для розшифрування. Ці процедури, в свою чергу, складаються з дрібніших елементів: для процедури обробки даних це етапи початкової модифікації даних, основної обробки даних, кінцевої модифікації даних; для процедури обчислення розпису ключа – етапи початкової модифікації ключа, основної обробки ключа, кінцевої модифікації підключів (рис. 4).

Розглянемо порядок обробки даних та ключа згідно з запропонованою узагальненою структурною організацією АСБШ.

Перед обробкою блоку вхідних даних обчислюється розпис ключа, результатом якого є набори підключів, що використовуються для шифрування даних. Якщо алгоритм шифрування задовольняє принципам Кергофа [11] та є абсолютно стійким, то розмір ключа шифрування більший за розмір або рівний розміру блоку вхідних даних. Рівність розмірів ключа та блоку даних можлива лише у випадку рівної ймовірності появи усіх значень ключа, тобто значення бітів ключа володіють однаковою ймовірністю появи та статистично незалежні один від одного. Однак для процедури обробки даних сумарний розмір підключів значно перевищує розмір ключа шифрування. Тому для отримання значної кількості підключів застосовується процедура обчислення розпису ключа, за допомогою якої будується набір підключів необхідного розміру.

Представимо організацію процедури обчислення розпису ключа як набір трьох взаємопов'язаних етапів (деякі з них можуть бути відсутні): початкової модифікації ключа, процедури основної обробки ключа та

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

кінцевої модифікації набору підключів (рис. 4).

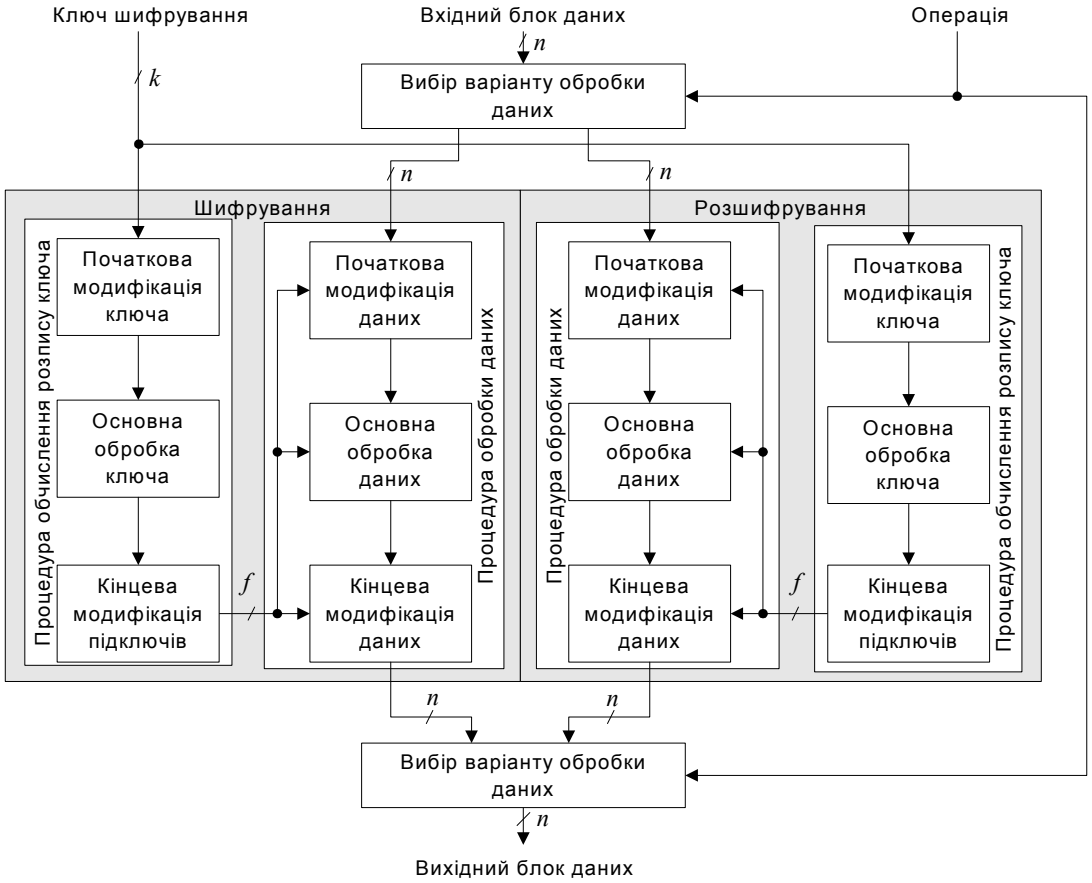

Рисунок 4 – Узагальнена структурна організація АСБШ

Етап початкової модифікації ключа призначений для формування початкового представлення вхідного ключа для наступного етапу та може включати в себе формування масивів даних, наприклад [14], вибір певних бітів вхідного ключа (наприклад [43, 44]) тощо. Обчислення етапу початкової модифікації підключа здійснюються як правило за однією й тією ж схемою, яка є фіксованою для заданого алгоритму шифрування. Крім цього, етап початкової модифікації ключа може здійснювати модифікацію виконуваних операцій процедур виконання етапів основної та кінцевої обробки ключа, процедури обробки даних відносно ключа та операції шифрування (наприклад, [46]). Модифікація може стосуватися як типу виконуваних операцій, так і даних, які додатково використовуються.

Етап основної обробки ключа має на меті формування заданого набору підключів для їх наступного використання в процедурі обробки даних. При цьому здійснюються різноманітні операції над деякими частинами вхідного ключа. Основною характеристикою процедури формування підключів є забезпечення впливу усіх бітів вхідного ключа на кожний підключ [47]. При цьому бажаною властивістю процедури є складність відновлення усього ключа шифрування за заданим підключем. Структура процедури залежить від виконуваної операції шифрування та алгоритму шифрування.

Етап кінцевої модифікації набору підключів призначений для переформування набору підключів для його наступного використання в процедурі обробки даних. Як і етап початкової модифікації ключа, цей етап може включати в себе перестановки елементів підключів (наприклад [44]), бітові маніпуляції (наприклад [46]) тощо. Структура процедури кінцевої модифікації набору підключів може бути різною для операцій шифрування та розшифрування, а конкретний її вид визначається описом алгоритму шифрування.

Обробка блоків вхідних даних здійснюється процедурою обробки даних, аргументами якої є вхідний блок та набір підключів. Кількість підключів та їх розмір є, як правило, фіксованими для певного алгоритму шифрування при шифруванні та розшифруванні вхідних блоків. У загальному випадку процедура обробки даних складається з трьох етапів (наприклад [14, 27, 40, 44, 46]): етапу початкової модифікації даних, етапу

основної обробки даних та етапу кінцевої модифікації даних. Кожний з етапів може використовувати для модифікації даних підключі, сформовані при обчисленні розпису ключа. Також деякі етапи можуть бути відсутніми.

Етап початкової модифікації вхідних даних призначений для їх перетворення у необхідний формат для подальшої обробки. Це перетворення може включати в себе як маніпуляції лише із самими даними (наприклад [44]), так і використання для їх модифікації елементів ключа шифрування (наприклад [27, 46]).

Етап основної обробки даних призначений для безпосереднього виконання раундів алгоритму шифрування, які є, як правило, подібними. При цьому для кожного раунду використовується свій набір раундових підключів, сформованих процедурою обчислення розпису ключа.

Завершує процедуру обробки даних етап кінцевої модифікації вихідного блоку, суть якої аналогічна до процедури початкової модифікації даних.

Запропонована структурна організація узагальненого АСБШ, функціональний та структурний поділ АСБШ на самостійні процедури дозволяє дослідити особливості організації обчислювального процесу АСБШ.

IV Організація обчислень узагальненого алгоритму симетричного блокового шифрування

Організація обчислення розпису ключа АСБШ

У загальному випадку процес обчислення розпису ключа та формування набору підключів можна представити як послідовність виконання кроків: отримати ключ шифрування, вибрати алгоритм обчислення розпису згідно з заданою операцією шифрування, обчислити та сформувати набір підключів. Розмір ключа шифрування є значно меншим за розмір необхідного набору підключів, тому процедури обчислення розпису ключа АСБШ використовують додаткові обчислення, причому для обчислення одного підключа може бути використаний як весь, так і деякі частини ключа шифрування, наперед визначені константи тощо. Однак, сучасні АСБШ можна умовно поділити на два класи – такі, які набір підключів отримують прямо із ключа шифрування, без використання додаткових обчислень (наприклад, [43]), та такі, які формують набір підключів, виконуючи додаткові обчислення (наприклад [44, 45, 46]). Ці два класи алгоритмів зведемо до одного класу алгоритмів із використанням додаткових обчислень, прийнявши, що для алгоритмів першого класу обчислення зводяться до формування підключів як підмножини ключа.

Аналіз особливостей структурної організації процедур формування наборів підключів АСБШ дозволив зробити висновок про можливість поділу цих процедур на такі типи: пряму, ітеративну та комбіновану. Наведемо основні характеристики, притаманні кожному варіанту виконання процедури обчислення розпису ключа.

Пряма процедура формування наборів підключів

Пряма процедура формування набору підключів передбачає використання лише елементів ключа та додаткової інформації, наприклад, спеціально обраних констант, які залежать від номера генерованого набору підключів. Прикладами АСБШ, що використовують пряму процедуру отримання наборів раундових підключів, є [40, 41, 43]. Таку процедуру формування деякого набору підключів $\{Sk\}_i$ із ключа шифрування можна представити у вигляді формули:

$$\{Sk\}_i = SFF_m\left(SFSK_m\left(SFI_m(K), i\right), i\right), \tag{7}$$

де $i = 0, \ldots, Nr + 1$, Nr – кількість наборів підключів, що відповідає сумі кількості наборів для етапу основної обробки даних та двох наборів для етапів початкової та кінцевої модифікації даних, K – ключ шифрування, SFF_m – функції кінцевої модифікації набору підключів при виконанні операції шифрування m, $SFSK_m$ – функції формування набору підключів при виконанні операції шифрування m, SFI_m – функції початкової модифікації ключа шифрування при виконанні операції шифрування m, $m = \{e, d\}$, e – операція шифрування, d – операція розшифрування, $Sk = \{sk_1, \ldots, sk_{Nsk}\}$, sk_j – j-й підключ, $j = 1, \ldots, Nsk$, Nsk – кількість підключів у наборі.

Структуру такої процедури (рис. 5) представимо як набір із процедури попередньої обробки ключа та паралельних процедур формування наборів раундових ключів.

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

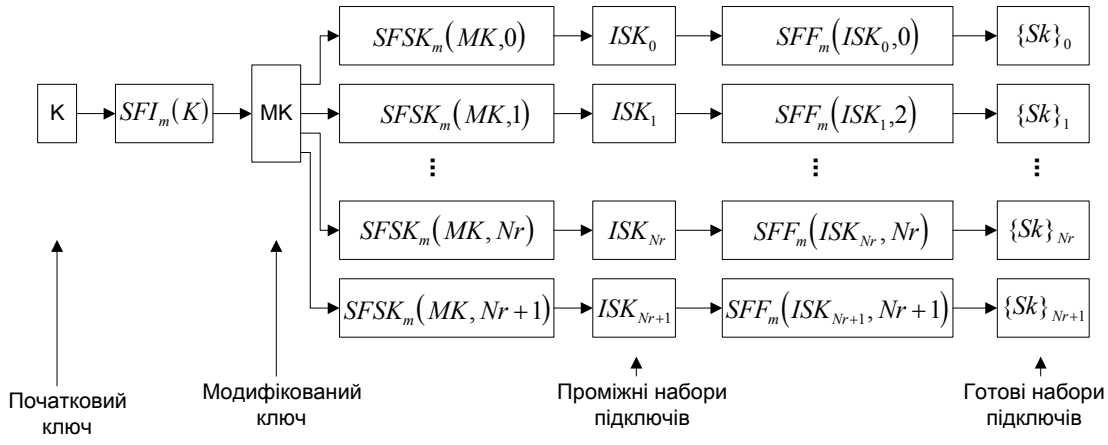

Рисунок 5 – Схема структурної організації прямої процедури обчислення розпису ключа

Кожною із паралельних процедур використовується весь або частина попередньо обробленого ключа шифрування MK, деяка додаткова інформація. Функції обчислення наборів підключів $SFSK_m$ формують проміжні набори підключів ISK_i, i = 0, …, Nr + 1, з яких функції кінцевої модифікації набору підключів формують набори підключів, що використовуються у процедурі обробки даних.

Ітеративна процедура формування наборів підключів

Ітеративна процедура передбачає використання ключа шифрування лише один раз для формування першого проміжного набору підключів (який, однак, також може використовуватися у процедурі обробки даних АСБШ) та подальшого ітераційного обчислення наступних наборів підключів, де проміжний набір використовується як початковий вхідний ключ. При обчисленнях може використовуватись додаткова інформація, аналогічно до прямої процедури. Прикладами АСБШ, що використовують ітеративну процедуру отримання наборів раундових підключів, є [12, 44, 46]. Ітеративну процедуру формування набору підключів представимо у вигляді формул:

$$\{Sk\}_0 = IFF^0{}_m\left(IFSK^0{}_m\left(IFI_m(K)\right)\right), \tag{8}$$

$$\{Sk\}_i = IFF^1{}_m\left(IFSK^1{}_m\left(\{Sk\}_{i-1}, i\right), i\right), \tag{9}$$

$$\{Sk\}_{Nr+1} = IFF^2{}_m\left(IFSK^2{}_m\left(\{Sk\}_{Nr}, Nr\right), Nr\right), \tag{10}$$

де $IFSK^0{}_m$, $IFSK^1{}_m$, $IFSK^2{}_m$ – функції обчислення набору підключів для початкової модифікації даних, наборів підключів для основного етапу обробки даних та набору підключів для кінцевої модифікації даних відповідно при виконанні операції шифрування m, IFF_m – функції кінцевої модифікації набору підключів при виконанні операції шифрування m, IFI_m – функції початкової модифікації ключа шифрування при виконанні операції шифрування m, $m = \{e,d\}$, e – операція шифрування, d – операція розшифрування.

Формула (8) використовується для обчислення набору підключів для початкової модифікації даних при $i = 0$, формула (9) використовується для послідовного обчислення наборів підключів при i = 1, …, Nr, формула (10) використовується для обчислення набору підключів для кінцевої модифікації даних при $i = Nr + 1$. Зауважимо, що додатковим параметром функцій $IFSK^1{}_m$, $IFSK^2{}_m$ можуть виступати відповідні проміжні набори підключів до їх модифікації функціями IFF^0_m, IFF^1_m, IFF^2_m.

Структуру такої процедури (рис. 6) представимо як набір із процедури попередньої обробки ключа та послідовних процедур формування наборів раундових ключів.

На рис. 6 пунктирними лініями показано варіант використання проміжних наборів підключів у функціях $IFSK^1{}_m$, $IFSK^2{}_m$.

Комбінована процедура формування наборів підключів

Комбінована процедура передбачає комбінування підходів прямої та ітераційної процедур: біжучий набір

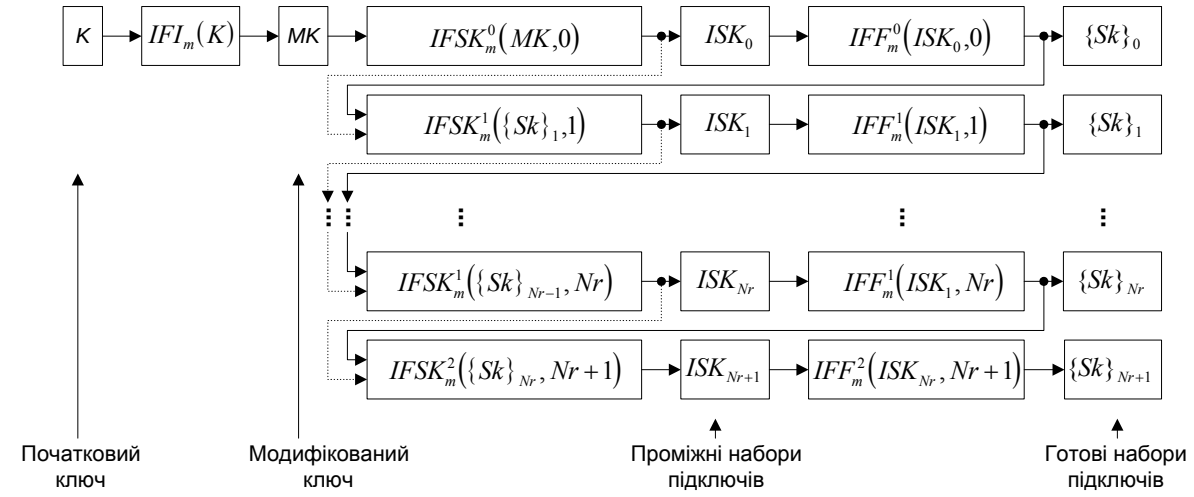

Рисунок 6 – Схема структурної організації ітеративної процедури обчислення розпису ключа

підключів обраховується із використанням як ключа шифрування, так і попереднього набору підключів (разом із додатковою інформацією). Прикладами АСБШ, що використовують ітеративну процедуру отримання наборів раундових підключів, є [35, 45, 48]. Комбіновану процедуру формування наборів підключів представимо у вигляді формул:

$$\{Sk\}_0 = CFF^0{}_m\left(CFSK^0{}_m\left(CFI_m(K)\right),0\right), \tag{11}$$

$$\{Sk\}_i = CFF^1{}_m\left(CFSK^1{}_m\left(\{Sk\}_{i-1},i\right),CFI_m(K),i\right), \tag{12}$$

$$\{Sk\}_{Nr+1} = CFF^2{}_m\left(CFSK^2{}_m\left(\{Sk\}_{Nr},Nr+1\right),CFI_m(K),Nr+1\right), \tag{13}$$

де $CFSK^0{}_m$, $CFSK^1{}_m$, $CFSK^2{}_m$ – функції обчислення набору підключів для початкової модифікації даних, наборів підключів для основного етапу обробки даних та набору підключів для кінцевої модифікації даних відповідно при виконанні операції шифрування m, CFF_m – функції кінцевої модифікації набору підключів при виконанні операції шифрування m, CFI_m – функції початкової модифікації ключа шифрування при виконанні операції шифрування m, $m = \{e,d\}$, e – операція шифрування, d – операція розшифрування.

Формула (11) використовується для обчислення набору підключів для початкової модифікації даних при $i = 0$, формула (12) використовується для послідовного обчислення наборів підключів при i = 1, …, Nr, формула (13) використовується для обчислення набору підключів для кінцевої модифікації даних при $i = Nr + 1$. Зауважимо, що додатковим параметром функцій $CFSK^1{}_m$, $CFSK^2{}_m$ можуть виступати відповідні проміжні набори підключів до їх модифікації функціями CFF_m^0, CFF_m^1, CFF_m^2.

Структуру такої процедури (рис. 7) представимо як комбінацію прямої та ітераційної структур.

Необхідно зазначити, що функції усіх етапів обробки для обчислення наборів підключів можуть різнитися між собою не тільки для операцій розшифрування та шифрування, а й для різних наборів підключів: функція етапу основної обробки ключа для набору i може відрізнятися від цієї ж функції для набору j при виконанні операції шифрування чи розшифрування. Тому додатковим параметром кожної функції є номер набору, для якого вона використовується.

Розглянувши три варіанти структурної організації обчислення наборів підключів для процедури обробки даних, перейдемо до розгляду організації обчислювального процесу при виконанні процедури обробки даних.

Організація обробки даних АСБШ

Процес обробки даних АСБШ представимо як послідовність таких етапів:

Етап 1: згідно з операцією шифрування та ключа шифрування провести налаштування структури та параметрів процедури обробки даних.

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

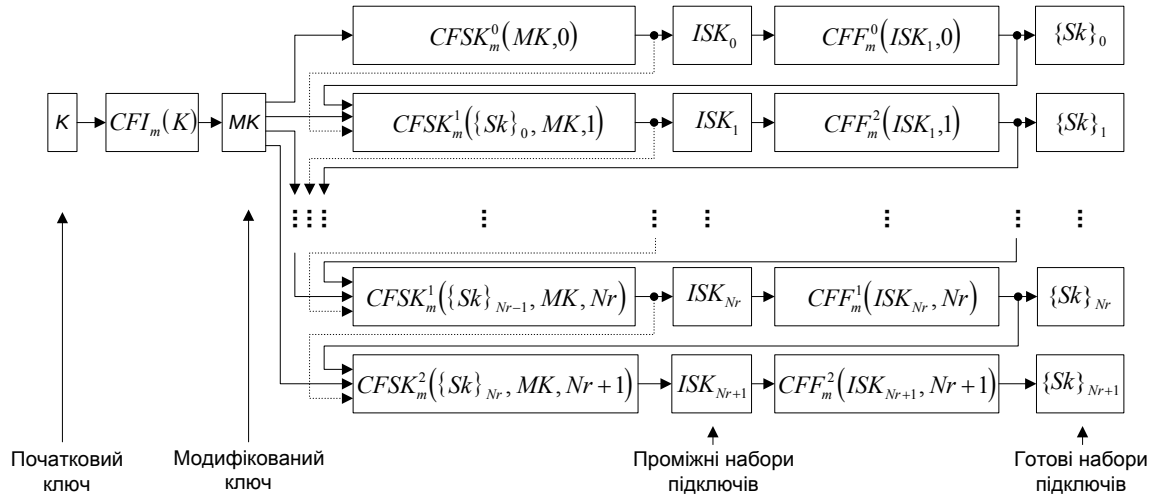

Рисунок 7 – Схема структурної організації комбінованої процедури обчислення розпису ключа

Етап 2: використовуючи набір підключів для етапу початкової модифікації даних обчислити модифікований блок даних.

Етап 3: використовуючи набори підключів для етапу основної обробки даних обчислити останній проміжний блок даних.

Етап 4: використовуючи набір підключів для етапу кінцевої модифікації даних обчислити вихідний блок даних.

Етап 1 виконується лише один раз при зміні ключа шифрування чи типу операції шифрування. Наступні етапи виконуються кожного разу при отриманні на вхід процедури обробки даних нового блоку даних.

Оскільки сучасні АСБШ володіють ітераційною структурною організацією процедури обробки даних [12 – 35, 43 – 46, 48] незалежною від типу операцій, які використовуються для обробки даних (фіксовані чи недетерміновані), то аналітично процес обробки даних представимо у вигляді формул:

$$IDM = IFD_m\left(ID, \{Sk\}_0\right), \tag{14}$$

$$IDI_1 = FD_m\left(IDM, \{Sk\}_1, 1\right), \tag{15}$$

$$IDI_i = FD_m\left(IDI_{i-1}, \{Sk\}_i, i\right), \tag{16}$$

$$OD = FFD_m\left(IDI_{Nr}, \{Sk\}_{Nr+1}\right), \tag{17}$$

де IDM – модифікований вхідний блок даних, IFD_m – функції виконання початкової модифікації даних при виконанні операції шифрування m, $m = \{e, d\}$, e – операція шифрування, d – операція розшифрування, $\{Sk\}_0$ – набір підключів для початкової модифікації даних, ID – вхідний блок даних, IDI_i – проміжний результат обробки блоку даних у i -му раунді, FD_m – функції виконання основного етапу обробки даних при виконанні операції шифрування m, $\{Sk\}_i$ – набір підключів для i -го раунду, i = 1, …, Nr, OD – вихідний блок даних, FDD_m – функції виконання кінцевої модифікації даних при виконанні операції шифрування m, $\{Sk\}_{Nr+1}$ – набір підключів для кінцевої модифікації даних.

Графічно процедуру обробки даних зобразимо як послідовність виконання деяких операцій над вхідним, проміжними та вихідним блоками даних (рис. 8).

Кожен етап процедури використовує свій набір підключів. Набори $\{Sk\}_i$, i = 1, …, Nr, використовуються для етапу основної обробки даних, $\{Sk\}_0$, $\{Sk\}_{Nr+1}$ – для початкової та кінцевої модифікації даних відповідно. Усі набори підключів обчислюються за допомогою процедури обчислення розпису ключа.

Аналогічно до процедури обчислення розпису ключа, функції усіх етапів обробки даних можуть різнитися між собою не тільки для операцій розшифрування та шифрування, а й для різних номерів раундів, як, наприклад, в [46]. Тобто функція етапу основної обробки даних для раунду i може відрізнятися від цієї ж

функції для набору j при виконанні операції шифрування чи розшифрування. Тому додатковим параметром кожної функції є номер раунду, для якого вона використовується.

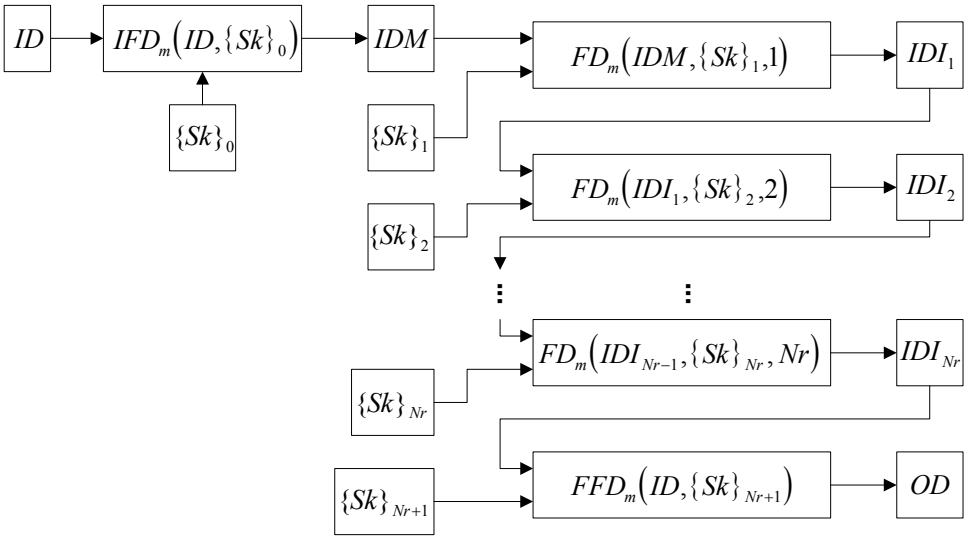

Рисунок 8 – Схема структурної організація процедури обробки даних

Отже, процедура обробки даних АСБШ має ітераційний характер, що дозволяє дослідити можливі варіанти загальної організації обчислень при обробці даних із різними типами процедур обчислення розпису ключа.

Загальна організація обчислень АСБШ

Так як до узагальненої структури АСБШ входять дві структурно самостійних та функціонально пов'язаних процедури – процедура обробки даних та процедура обробки ключа, то виділяються два способи їх виконання відносно одна одної: одноразове та паралельне (рис. 9).

Процес формування масиву підключів можна проводити лише один раз при використанні нового ключа шифрування (одноразове виконання) та паралельно із обробкою даних (паралельне виконання).

У першому випадку при обробці блоку даних обчислені набори підключів зберігаються у деякій пам'яті та використовуються в міру необхідності. Подальший процес обробки наступних блоків даних передбачає використання вже готових наборів. При зміні операції чи ключа шифрування ініціюється процедура обчислення розпису ключа та проводиться формування та збереження усіх наборів підключів із нового ключа. Якщо процедура обробки даних використовує додаткові модифікації блоку даних перед та після виконання етапу основної обробки даних, то необхідні для цього дані також обчислюються при виконанні процедури обчислення розпису ключа.

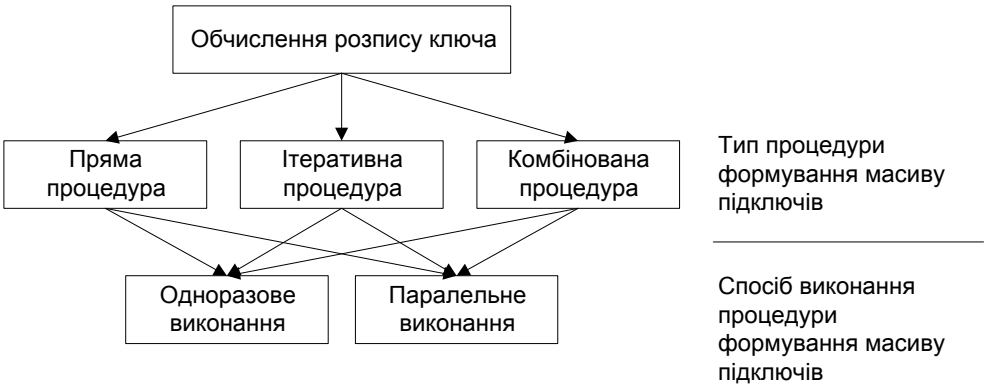

Рисунок 9 – Класифікація способів організації обчислень розпису ключа АСБШ

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

У другому випадку необхідний набір підключів формується паралельно із виконанням процедури обробки даних, тобто “на льоту”. Використання певного обчисленого набору підключів проводиться на відповідному етапі виконання процедури обробки даних. Випадок паралельного обчислення набору підключів передбачає паралельне виконання двох процедур – обробки даних та формування розпису ключа. Зміна ключа чи операції шифрування приводить до необхідності ініціації налаштування структур та параметрів процедур обчислення розпису ключа та обробки даних.

Вибір конкретного способу залежить від структури процедури обчислення розпису ключа, структури процедури обробки даних, організації обчислень з обробки даних та області застосування засобу, який реалізує АСБШ. Переваги та недоліки кожного способу організації обчислень можна встановити лише у випадку заданих обмежень на реалізацію АСБШ, характеру обробки даних (частотою зміни ключа та операції шифрування), частотою надходження даних тощо.

V Висновки

У роботі досліджено базові підходи та особливості структурної організації сучасних АСБШ. Досліджено та розроблено структурну організацію узагальненого АСБШ, де виділено такі основні структурні елементи як процедуру обробки даних та процедуру обчислення розпису ключа, які в свою чергу складаються з: процедур обробки даних для операцій шифрування та розшифрування, процедур обчислення розпису ключа для операцій шифрування та розшифрування. Для кожної з перелічених процедур запропоновано поділ на складові етапи: початкової модифікації, основної обробки та кінцевої модифікації даних чи набору підключів, що дозволило дослідити особливості структурної організації процедур формування наборів підключів та запропонувати класифікацію типів цих процедур (пряму, ітеративну та комбіновану), побудувати для кожного типу процедури аналітичні вирази, запропонувати графічні інтерпретації структур процедур.

Встановлено, що процедура обробки даних АСБШ має ітераційний характер. Використовуючи запропонований функціональний та структурний поділ АСБШ на самостійні процедури, описані основні підходи виконання складових АСБШ, на основі яких проведено класифікацію способів його виконання та виділено два таких варіанти виконання формування масиву підключів відносно виконання процедури обробки даних: лише один раз при використанні нового ключа шифрування та паралельно із обробкою даних.

Література: 1. В. А. Пичуев, А. Г. Рябченко, Д. Г. Титов, С. А. Фролов "О проектировании СБИС высокоскоростного криптопроцессора" // Автометрия № 6, 1994. С. 91 – 98. 2. Мельник А. О. "Спеціалізовані комп'ютерні системи реального часу". – Львів: ДУ "Львівська політехніка", 1996. – 53 с. 3. Davio M., Desmedt Y., Fosseprez M., Govaerts R., Hulsbosch J., Neutjens P., Piret P., Quisquater J. J., Vandewalle J. and Wouters P. "Analytical characteristics of DES" //Advances in Cryptology – CRYPTO-83 Proceedings, New York, NY: Springer-Verlag, pp. 171 - 202. 4. А. Мельник, Т. Коркішко "Стан та напрямки розвитку надвеликих інтегрованих схем захисту інформації" //Збірник праць Другої науково-технічної конференції "Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні", Київ 24 - 26 квітня 2000, с. 275 - 281. 5. T. Korkishko, V. Melnyk "DES Cryptographic Processor" Report on the research project. Georg-Simon-Ohm-Fachhochschule Nuernberg 27 September 1999 – 28 November 1999, 178 p. 6. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone “Handbook of Applied Cryptography”, CRC Press, October 1996, 816 p. 7. Шеннон К. Э. "Теория связи в секретных системах" // В кн.: Шеннон К. Э. Работы по теории информации и кибернетике. М.:, Изд. иностр. лит., 1963. С. 333 – 402. 8. B. Schneier "Applied Cryptography", John Wiley & Sons, 1996. 9. E. Biham, A. Shamir, "Differencial Cryptanalysis of the Data Encryption Standard", Springer-Verlag, 1993. 10. M. Matsui, "Linear Cryptanalysis method for DES cipher". Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765, T. Helleseth, Ed., Springer-Verlag, 1994, pp. 386 - 397. 11. Молдовян А. А., Молдовян Н. А., Советов Б. Я. Криптография. – Серия "Учебники для вузов. Специальная литература". – СПб.: Издательство "Лань", 2000. – 2 24 с., илл. 12. Daemen, R. Govaerts, and J. Vandewalle, "A New Approach to Block Cipher Design," Fast Software Encryption, Cam bridge Security Workshop Proceedings, Springer-Verlag, 1994, pp. 18 - 32. 13. Anderson AND E. Biham, "Two practical and provably secure block ciphers: BEAR and LION", D. Gollmann, editor, Fast Software Encryption, Third International Workshop (LNCS 1039), 113 – 120, Springer-Verlag, 1996. 14. B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)", Fast Software Encryption, LNCS 809, R. Anderson, Ed., Springer-Verlad, 1994, pp. 191 - 204. 15. C. M. Adams, "Constucting symmetric ciphers using CAST design procedure", Designs, Codes, and Cryptography, Vol. 12, No. 3, Movember 1997, pp. 71 - 104. 16. Shimizu and S. Miyaguchi. "Fast data encipherment algorithm FEAL". In Advances in Cryptology Eurocrypt ’87, pages 267 - 280, Springer-Verlag, 1988. 17. Merkle, "Fast Software

Encryption Functions", Advances in Cryptology CRYPTO'90, Proceedings, Springer-Verlag, 1991, pp. 476 - 501. 18. L. Brown, J. Pieprzyk, and J. Seberry, "LOKI – a cryptographic primitive for authen-tication and secrecy applications", Advances in Cryptology–AUSCRYPT '90 (LNCS 453), 229 – 236, 1990. 19. Brown, M. Kwan, J. Pieprzyk, and J. Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI, " Advances in Cryptology ASIACRYPT '91 Proceedings, Springer-Verlag, 1993, pp. 36 - 50. 20. Smith, "The design of Lucifer: A cryptographic device for data communications", IBM Research Report RC 3326, IBM T. J. Watson Research Center, Yorktown Heights, N.Y., 10598, U.S.A., Apr. 15 1971. 21. Blaze and B. Schneier. "The MacGuffin Block Cipher Algorithm". Fast Software Encryption, Second International Workshop Proceedings (December 1994), Springer-Verlag, 1995, pp. 97 - 110. 22. C. Burkwick, D. Coppersmith, E. D'Avignon at all, "MARS – a candidate cipher for AES" in First Advanced Encryption Standard (AES) Conference, Ventura, C A, 1998. 23. Daemcn, R. Govacrts, and J. Vandewalle, "Block Ciphers Based on Modular Arithmetic". Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography, Rome, Italy, 15 - 16 Feb 1993, pp. 418. 80 - 89. 24. M. Matsui, "New block encryption algorithm MISTY". Fast Software Encryption, LNCS 1267, E. Biham, Ed., Springer-Verlag, 1997, pp. 54-68. 25. Connell, "An Analysis of NewDES: A Modified Version of DES". Cryptologia, v. 14, n. 3, Jul 1990, pp. 217-223. 26. Robshaw, "Block Ciphers," Technical Report TR-601, RSA Laboratories, Jul 1994. 27. L. Rivest, "The RC5 encryption algorithm", B. Preneel, editor, Fast Software Encryption, Second International Workshop (LNCS 1008), 86–96, Springer-Verlag, 1995.28. R. Rivest, M. Robshaw, R. Sidney, and Y.Yin, "The RC6 Block Cipher", in First Advanced Encryption Standard(AES) Conference, Ventura, CA, 1998. 29. Cusick and M.C. Wood, "The REDOC-II Cryptosystem," Advances in Cryptology CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 545-563. 30. Massey. "SAFER K-64: A byte-oriented block ciphering algorithm". In Proceedings of 1st International Workshop on Fast Software Encryption, p. 1-17, Springer-Verlag, 1993. 31. Massey. "SAFER K-64: One year later". In Proceedings of 2nd Workshop on Fast Software Encryption, p. 212-241, Springer-Verlag, 1995. 32. R. Anderson, E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard", In First Advanced Encryption Standard (AES) Conference, (Ventura, CA), 1998. 33. J. Daemen, L. R. Knudsen, V. Rijmen "The block cipher Square", Fast Software Encryption, LNCS 1267, E.Biham, Ed., Springer-Verlad, 1997, pp. 149-165. 34. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm", B. Pre-neel, editor, Fast Software Encryption, Second International Workshop (LNCS 1008), 363– 366, Springer-Verlag, 1995. 35. B. Schneier, J. Kesey, D. Whiting, D. Wagner, and C.Hall, "Twofish: A 128-bit Block Cipher", In First Advanced Encryption Standard (AES) Conference, (Ventura, CA), 1998. 36. Feistel H. Cryptography and computer privacy. Sci.Am., May 1973, vol. 228(5), p. 15-23. 37. B. Schneier, J. Kelsey "Unbalanced Feistel Networks and block cipher design", FastSoftware Encryption, LNCS 1039, D.Gollman, Ed., Springer-Verlag, 1996, pp. 121-144. 38. Moldovyan N. A., Moldovyan A. A., Zaikin O. A., Undetermined Software-Oriented Ciphers Based on Data-Depended Subkey Selection // Int. Conf. Computer Methods in Control Systems CMCS'97. December 11-12, 1997. Szczecin, Poland. Proc. pp. 157-164. 39. Молдовян Н. А., Молдовян А. А., Алексеев Л. Е. Перспективы разработки скоростных шифров на основе управляемых перестановок // Вопросы защиты информации, 1999, № 1, с. 41-47. 40. Молдовян А. А., Молдовян Н. А. Способ криптографического преобразования блоков двоичных данных. Патент РФ № 2 141729. МПК[6] H04 L 9/00. Бюл. N 32 от 20. 11. 99. 41. Грушвицкий Р. И., Савлуков Н .В. Скоростные устройства шифрования на базе нового криптографического примитива // В кн. Межрегиональная конференция "Информационная безопасность регионов России ИБРР-99". Санкт-Петербург, 13-15 октября 1999 г. тезисы конф. Ч. 1. С. 47-48. 42. Н. А. Молдовян. "Скоростные блочные шифры". – СПб, Издательство СПбГУ, 1998. – 230 с. 43. ГОСТ28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М. Госстандарт СССР. 44. FIPS 46, "Data Encryption Standard", Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D. C. 45. X. Lai, J. L. Massey A proposal for a New Block Encryption Standard. Advances in Cryptology – EUROCRYPT-90 Proceedings, New York, NY: Springer-Verlag, pp. 389-404. 46. J. Daemen and V. Rijmen. AES Proposal: Rijndael. In First Advanced Encryption Standard(AES) Conference, Ventura, CA, 1998. 47. J. Kelsey, B. Schneier, D. Wagner, "Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Tripple-DES", Advances in Cryptology, Proceedings Crypto'96, LNCS 1109, N. Koblitz, Ed., Springer-Verlag, 1996, pp. 237-252. 48. L. Knudsen, "DEAL - A 128-bit Block Cipher", in First Advanced Encryption Standard (AES) Conference, Ventura, CA, 1998.