Design of block cipher (FBC) based on the Feistel network

Oleksandr Polischuk, Andriy Lagun

Security of Information Technologies Department, Lviv Polytechnic National University, UKRAINE,
Lviv, S. Bandery street 12,
E-mail: AYPolishchuk @gmail.com
E-mail: Lagun_ae@polynet.lviv.ua

This paper describes a block cipher designed by Feistel network, investigates the cryptographic properties of cipher and round key generation algorithm compared with primitive cryptographic operations, and tested by DIEHARD tests.

Encryption and decryption of information by this algorithm is described in the formulas (1) and (2), structural diagram of the encryption algorithm is shown in Fig.1.

This algorithm is a set of combined transformations of two types, one of those aims at dispersion bits around the block of text, and another bit of dispersion in only one part of it.

These transformations and reverse ones are represented in the formulas (3) - (10). Complex functions are used in data transformations represented in the formulas (11) - (17). Structure of complex functions F1 and F2 are shown in Fig.2 and Fig.3, respectively.

For the formation of round keys input key value is modified by an XOR operation on its parts and by a round key that has a principle of work similar to Galois register, but without shifting. This algorithm is presented in the formula (18), its structural diagram is shown in Fig.4.

Results comparing this algorithm with primitive cryptographic functions is shown in Fig.5. Results of round key generation algorithm investigation are presented in Table 1. Results of testing the algorithm on a set of DIEHARD tests are presented in Table 2.

Although not all tests were completed successfully, but this algorithm should be considered as a prototype of a block cipher that requires further research and improvement.

Whereas the basic requirements of the design were flexibility and the ability to easy replacement of its components without violating the basic structure of the system, this code can be easily improved or modified.

Проектування блокового шифру (ФБШ) на основі перетворення Фейстеля

Олександр Поліщук, Андрій Лагун

Кафедра безпеки інформаційних технологій,
Національний університет “Львівська політехніка”,
УКРАЇНА, м.Львів, вул.С.Бандери, 12,
E-mail: AYPolishchuk@gmail.com,
E-mail: Lagun_ae@polynet.lviv.ua

В даній роботі здійснено опис спроектованого на основі перетворення Фейстеля блокового шифру, досліджено криптографічні властивості даного шифру та алгоритму генерування раундових ключів у порівнянні з примітивними криптографічними операціями, проведено тестування даної системи з допомогою тестів DIEHARD.

Ключові слова – криптографія, блоковий шифр, перетворення Фейстеля, проектування криптографічних алгоритмів, дослідження криптостійкості.

I. Вступ

В зв’язку з швидкими темпами розвитку інформаційних систем та різким збільшенням об’ємів інформації, що обробляється, питання захисту інформації стає ще більш актуальним. Одним з найпоширеніших способів, що дозволяє забезпечити всі аспекти захисту інформації, є шифрування.

При наявності належного каналу передачі секретного ключа, симетричні алгоритми мають ряд істотних переваг, зокрема це простота та висока швидкодія [1].

Найпоширенішим типом симетричних шифрів, є блокові шифри. Незважаючи на те, що існує велика кількість блокових шифрів, більшість з них мають ряд недоліків або є недостатньо стійкими. Тому основною метою даної роботи є розробка та дослідження блокового шифру на основі перетворення Фейстеля.

II. Структура алгоритму.

Даний криптографічний алгоритм використовує [2] прямі перетворення Фейстеля при зашифруванні інформації та зворотні – при розшифруванні. Структуру схеми шифрування даного алгоритму представлено на Рис.1.

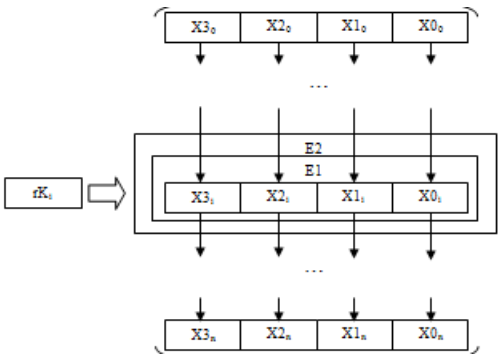

Рис.1. Загальна структура схеми шифрування алгоритму.

Зашифрування та розшифрування інформації даним алгоритмом в загальному описується формулами (1) та (2) відповідно.

$$X_i = E2(E1(X_{i-1}, rK_i), rK_i) \quad (1)$$

$$X_i = D1(D2(X_{i-1}, rK_{n+1-i}), rK_{n+1-i}), \quad (2)$$

де X – вхідна інформація, E1, E2, D1, D2 – функція зашифрування, розшифрування, відповідно, rK – раундовий ключ. В алгоритмі застосовуються різні типи перетворення Фейстеля з різними властивостями. Це дозволяє досягти різного рівня розсіювання та неявного використання перестановок різного типу в одному раунді.

При зашифруванні інформації, над кожним з блоків здійснюється пара перетворень E1 та E2, що чергуються між собою. При розшифруванні – використовується пара перетворень D1 та D2, що є зворотними до попередніх і, відповідно, повинні застосовуватися в зворотному порядку.

Перетворення E1 приймає на вхід блок інформації, що розбивається на чотири рівні частини $X0_{i-1}$ - $X3_{i-1}$. Частина $X3_{i-1}$ та раундового ключа rK_i, є вхідними аргументами функції F1, яка бере участь в утворенні трьох вихідних частин перетворення $X1_i$ – $X3_i$. Кожна з цих частини є результатом операції виключного АБО над одним з виходів функції F1 та однією з частин $X0_{i-1}$-$X2_{i-1}$. Частина $X0_i$ формується за рахунок циклічного зсуву частини $X3_{i-1}$ на і розрядів. Як бачимо дане перетворення забезпечує модифікацію усіх вхідних частин, шляхом складних та простих їх перетворень, причому залежність перетворення E1 від номера раунду, забезпечує щоразу дещо іншу модифікацію частин блоку. Дане перетворення описується формулами (3) та (4).

$$X0_i = X3_{i-1} << i \quad (3)$$

$$X(t)_i = F1(i, X3_{i-1}, rK_i)_1 \oplus X(t-1)_{i-1} \quad (4)$$

Для виконання зворотних дій застосовується перетворення D1, воно виконується за аналогічним принципом та має такі ж властивості, як і E1. Перетворення описуються формулами (5) та (6).

$$X(t)_i = F1(X0_{i-1} >> (n+1-i), rK_{n+1-i})_1 \oplus X(t+1)_{i-1} \quad (5)$$

$$X3_i = X0_{i-1} >> (n+1-i) \quad (6)$$

Перетворення E2 приймає на вхід блок інформації, що розбивається на чотири рівні частини $X0_{i-1}$-$X3_{i-1}$. Частини $X0_{i-1}$-$X2_{i-1}$ та раундовий ключ rK_i, є вхідними аргументами функції F2, яка формує вихідну частину перетворення $X0_i$, шляхом виконання операції виключного АБО над виходом функції F2 та частиною $X3_{i-1}$. Вихідні частини $X1_i$-$X3_i$ є не модифікованими частинами $X0_{i-1}$-$X2_{i-1}$, відповідно. Дане перетворення забезпечує модифікацію лише однієї вхідної частини, але ця модифікація залежить від усіх частин вхідного блоку $X0_{i-1}$-$X3_{i-1}$ та раундового ключа rK_i. Для виконання зворотних дій застосовується перетворення D2, яке виконується аналогічно та має такі ж властивості, як і E2. Перетворення E2 та D2 описуються формулами (7) – (10).

$$X0_i = F2(i, X2_{i-1}, X2_{i-1}, X0_{i-1}, rK_i) \oplus X3_{i-1} \quad (7)$$

$$X(t)_i = X(t-1)_{i-1} \quad (8)$$

$$X(t)_i = X(t+1)_{i-1} \quad (9)$$

$$X3_i = F2(i, X3_{i-1}, X2_{i-1}, X1_{i-1}, rK_{n+1-i}) \oplus X0_{i-1} \quad (10)$$

Вищеописані перетворення мають різні ступені розсіювання та містять неявні перестановки різного типу за рахунок використання операції циклічного зсуву та, власне, структури перетворень. Перетворення E1 та D1 орієнтовані на розсіювання біт усього вхідного блоку, та перестановок в одній з його частин. Перетворення E2 та D2 орієнтовані на розсіювання біт в одній з частин вхідного блоку, та перестановок в усьому блоці. Наявність методів розсіювання та перестановки, а також чергування перетворень різного типу забезпечує швидке досягнення «лавинного» ефекту. Крім наведеної вище, можлива інша послідовність застосування даних перетворень, при якій наступне не є оберненим до попереднього.

Функція F1, структура якої представлена на Рис.2, має три вхідних аргументи: номер раунду і, блок даних X та раундовий ключ rK розміром 64 біта; та повертає вихідні значення $F1_1$ – $F1_3$.

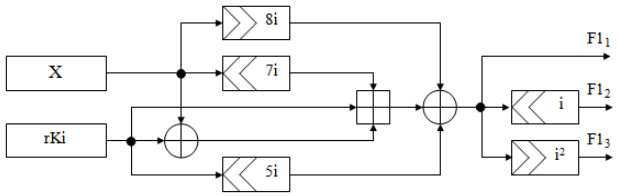

Рис.2. Структура функції складного перетворення F1.

Дана функція використовує операції трьох типів: виключне АБО, додавання за модулем 264 та циклічний зсув. Ці операції належать до ортогональних алгебраїчних груп і є несумісні, в тому сенсі, що вони не задовольняють дистрибутивний та асоціативний закони. Використання таких операцій забезпечує захист від лінійного та диференціального криптоаналізу, за умови їх чергування. Фактично, все перетворення, що виконує дана функція, надходить на вихід $F1_1$, а інші два виходи є різними перестановками даного результату, що досягаються за рахунок використання циклічних зсувів. Дана функція описується формулами (11) – (13).

$$F1_1(X1, rK_i) = ((rK_i \oplus X1) + (X1 << 7i) + rK_i) \oplus \oplus (X1 >> 8i) \oplus (rK_i << 5i) \quad (11)$$

$$F1_2(X1, rK_i) = F1_1(X1, rK_i) << i \quad (12)$$

$$F1_3(X1, rK_i) = F1_1(X1, rK_i) >> i^2 \quad (13)$$

Використання зсувів різного типу на різну кількість позицій, при проходженні достатньої кількості раундів забезпечує перестановку по всій множині біт вхідного блоку. Розсіювання біт вхідного блоку досягається шляхом накладання один на одного початкових та модифікованих вхідних аргументів. Застосування функцій, що базуються на даних перетвореннях дозволяє відмовитися від використання таблиць заміни, без втрати стійкості.

Функція F2, структура якої представлена на Рис.3, приймає п'ять вхідних аргументів: номер раунду i, блоки даних X1-X3 та раундовий ключ rK, розміром 64 біта; та повертає одне вихідне значення. На відміну від попередньої у даній функції використовуються три константи const1, const2 і const3, що забезпечують додаткове розсіювання біт. Дана функція, разом з деякими попередніми обчисленнями, представлена у формулах (14)–(17).

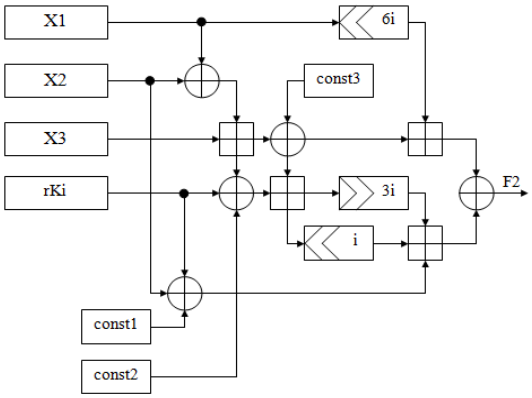

Рис.3. Структура функції складного перетворення F2.

$$l_1 = (X1 \oplus X2) + X3 \quad (14)$$

$$l_2 = l_1 \oplus const1 \quad (15)$$

$$l_3 = l_2 + (l_1 \oplus rK_i \oplus const2) \quad (16)$$

$$F2(X1, X2, X3, rK_i) = (X1 << 6i) + l_2) \oplus \oplus ((l_3 >> 3i) + (l_3 << i) + (X2 \oplus rK_i \oplus const3)) \quad (17)$$

Функція F2, застосовується для модифікації лише однієї частини, але дозволяє забезпечити більший ефект розсіювання, за рахунок складнішої внутрішньої структури. При реалізації алгоритму, константи const1-const3 можна ініціалізувати довільними статичними значеннями, або застосовувати, як додаткову ключову інформацію, що забезпечить вищу стійкість. Бажано, щоб значення констант було максимально випадковою послідовністю біт, що дасть змогу забезпечити вищу стійкість, ніж заповнення з певною періодичністю.

Крім того, що дана функція застосовується у перетвореннях E2 та D2, вона використовується для формування раундових ключів, що дозволяє формувати сильні підключі, навіть при поданні на вхід алгоритму ключа малого розміру, а константи const-1const3 дозволять сформувати раундовий ключ навіть при відсутності вхідного ключа, як такого. Також це дозволяє зменшити розмір програми і складність пристрою при реалізації даного алгоритму.

Розглянемо детальніше механізм формування раундових ключів. Вхідний ключ при необхідності доповнюється нульовими бітами до розміру 256 біт. Структурна схема формування ключа для i-го раунду зображена на Рис.4.

Функцію F2 представлено формулою (18). Вона приймає частини поточного ключа як вхідні аргументи в порядку, який визначається номером раунду. Вихідне значення є раундовим ключем.

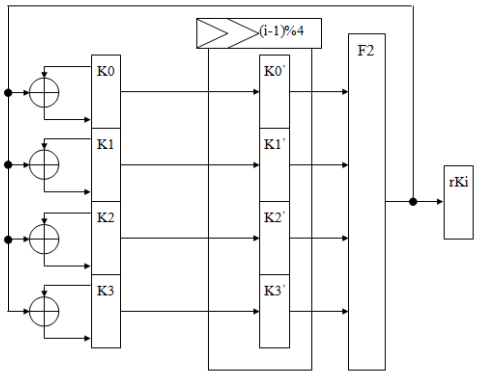

Рис.4. Механізм формування раундового ключа.

$$rK_i = F2(K((i-1)\%4),\ K(i\%4),\ K((i+1)\%4),\ K((i+2)\%4)) \quad (18)$$

Для формування сильних раундових ключів значення вхідного ключа модифікується за рахунок виконання операції виключного АБО над його частинами та раундовим ключем, що схожий за принципом роботи до регістру Галуа, але з відсутністю зсуву.

II. Дослідження криптографічних властивостей алгоритму

Розглянемо властивості механізму формування раундового ключа в залежності від поданої на вхід алгоритму ключової інформації. Для порівняння використаємо механізм формування раундового ключа, що використовується в спроектованому алгоритмі та механізм формування раундового ключа на основі операції виключного АБО. Отримані згенеровані ключі для трьох раундів представлено у Табл.1.

ТАБЛИЦЯ 1.
РЕЗУЛЬТАТИ РОБОТИ МЕХАНІЗМУ ГЕНЕРУВАННЯ КЛЮЧІВ

Вхідний ключ	Раундові ключі	
	На основі F2	На основі XOR
29:3d:61:ae:98:68:c6:5d:a3:c5:57:e5:3f:77:59:7e	f1:1a:40:f0:23:f0:80:24 12:da:b9:08:41:35:e1:ea 9d:4b:fc:7e:36:ca:db:41	8a:f8:36:4b:a7:1f:9f:23 8a:f8:36:4b:a7:1f:9f:23 8a:f8:36:4b:a7:1f:9f:23
6a:98:a8:a7:50:fb:e2:bf:77:5c:94:a6:f2:2d:0c:36:46:41:0a:31:3a:ca:11:45:96:bf:58:91:1f:c1:53:d0	c5:2d:c3:9d:84:e1:8d:65 45:d7:c8:95:14:98:d4:8f d4:f6:bc:ba:00:5d:cc:46	cd:3a:6e:a1:87:dd:ac:1c cd:3a:6e:a1:87:dd:ac:1c cd:3a:6e:a1:87:dd:ac:1c

Як бачимо при використанні механізму генерування раундового ключа на основі операції виключного АБО у випадку відсутності вхідного ключа або при певній його періодичності ми отримуємо нульові раундові ключі, а при наявності вхідного ключа різних розмірів ми отримуємо однакові прогнозовані раундові ключі. При використанні спроектованого механізму формування раундових ключів ми у будь-якому випадку отримуємо різні неперіодичні раундові ключі.Розглянемо

властивості перетворення Фейстеля, що використовуються при зашифруванні інформації з використанням різних функцій складного перетворення окремо та у поєднанні після одного раунду роботи. Використаємо спроектовані перетворення E1 та E2 з функціями складного перетворення F1 та F1 та порівняємо їх з результатами отриманими при застосуванні простих функцій, що базуються на виключному АБО. Результати зміни представлені у вигляді бітових матриць на Рис.5.

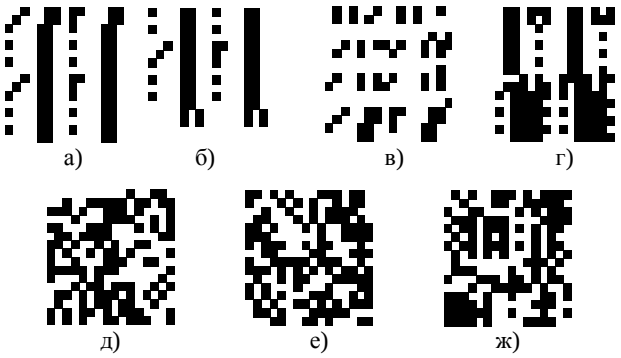

Рис.5. Бітові матриці перетворень: а) – початковий файл; б) – E1, XOR; в) – E2, XOR; г) – E1, E2, XOR; д) – E1, F1; е) – E2, F2; ж) – E1, E2, F1, F2

Як бачимо застосування перетворень на базі функцій F1 та F2 дозволяє досягти необхідних ефектів розсіювання біт випадковим чином. Якщо розглянути результати, отримані при застосуванні перетворень з використанням виключного АБО, то побачимо явні статистичні залежності.

Таким чином, алгоритм демонструє хороші криптографічні властивості на прикладі розсіювання біт випадковим чином по усьому об'єму блоку інформації.

Результати тестування алгоритму з допомогою набору тестів DIEHARD представлено у Табл.2.

З одержаних результатів тестування можна зробити наступні висновки: отримані значення p-value різних тестів є ймовірнісними величинами. Найбільші та найменші значення даної величини є ознакою того, що послідовність не є випадковою, зазвичай p-value для зарахування тесту як пройденого має знаходитися в діапазоні 0,01-0,99 [3].

З таблиці видно, що не всі тести були пройдені успішно, проте даний алгоритм варто розглядати, передусім, як прототип блочного шифру, що, безумовно, потребує подальшого дослідження та вдосконалення.

ТАБЛИЦЯ 2

Результати тестування алгоритму з використанням DIEHARD

Назва тесту	Значення p-value	
	Відкритий текст	32 раунди алгоритму
Birthday spacing test	1.000000	0.696130
Overlapping 5-permutation	1.000000	0.999962
Binary rank test for 31x31	1.000000	0.991145
Binary rank test for 32x32	1.000000	0.998648
Binary rank test for 6x8	1.000000	0.960185
Bitstream test	1.000000	0.896750
OPSO	1.000000	1.000000
OQSO	1.000000	1.000000
DNA	1.000000	0.999200
Count-the-1's on a stream	1.000000	0.999999
Count-the-1's for specific	1.000000	0.903037
Parking lot test	0.704165	0.114437
Minimum distance test	1.000000	1.000000
3Dspheres test	0.999976	0.811422
Squeeze test	0.999999	0.502241
Overlapping sums	0.992938	0.459717
Runs test	0.780865	0.232031
Craps test	0.953170	0.575279

Висновок

В даній роботі розглянуто питання розробки блокових систем шифрування та спроектовано блоковий алгоритм на основі схеми Фейстеля, що має різні типи перетворень в одному раунді, базується на простих логічних та арифметичних операціях, а також володіє механізмом формування раундових ключів. Наведено результати дослідження даного алгоритму за допомогою тестів DIEHARD та проведено порівняння криптографічних властивостей даного алгоритму з примітивними криптографічними перетвореннями. Одержані результати довели ефективність спроектованого блокового шифру.

Література

[1] Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М: Издательство Триумф, 2003.

[2] Баричев С. Г., Гончаров В. В., Серов Р. Е. Основы современной криптографии. – М.:Телеком, 2001.

[3] Instructions for using DIEHARD: a battery of tests of randomness. 1997.