­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­



Контрольна робота №2

Комп'ютерна криптографія. Мережа Фейселя.

з курсу «Технології захисту інформації»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 8

Виконав студент гр. КНз-3

Чалий Михайло

­­

Львів 2015

Зміст

Криптографія 3

Термінологія 3

Дія криптографії 3

Сучасний стан криптографії 4

Мережа Фейстеля 5

Історія 5

Конструкція блочного шифру на основі мереж Фейстеля 6

Шифрування 6

Алгоритмічний опис 7

Функції, використовувані в мережах Фейстеля 8

S-блок 9

P-блок 10

Циклічний зсув 10

Додавання по модулю n 11

*Множення по модулю n* 11

Переваги і недоліки 11

Теоретичні дослідження 12

Модифікації мережі Фейстеля 12

Шифри на основі мережі Фейстеля 13

Люцифер (Lucifer) 13

DES 15

ГОСТ 28147-89 16

Література 17

# Криптографія

**Криптогра́фія** (від грецького *kryptós* — прихований і *gráphein* — писати) — наука про математичні методи забезпечення конфіденційності (неможливості прочитання інформації стороннім) і автентичності (цілісності і справжності авторства) інформації. Розвинулась з практичної потреби передавати важливі відомості найнадійнішим чином. Для математичного аналізу криптографія використовує інструментарій абстрактної алгебри та теорії ймовірностей.

## Термінологія

Тривалий час під криптографією розумілось лише *шифрування* — процес перетворення звичайної інформації (*відкритого тексту*) в незрозуміле «сміття» (тобто, *шифротекст*).[1] Дешифрування — це зворотний процес відтворення інформації із шифротексту. Шифром називається пара алгоритмів шифрування/розшифрування. Дія шифру керується як алгоритмами, та, в кожному випадку, *ключем*. Ключ — це секретний параметр (в ідеалі, відомий лише двом сторонам) для окремого контексту під час передачі повідомлення. Ключі мають велику важливість, оскільки без змінних ключів алгоритми шифрування легко зламуються і непридатні для використання в більшості випадків. Історично склалось так, що шифри часто використовуються для шифрування та дешифрування, без виконання додаткових процедур, таких як аутенифікація або перевірка цілісності.

В англійській мові слова *криптографії* та *криптології* інколи мають однакове значення, в той час, як деколи під *криптографією* може розумітись використання та дослідження технологій шифрування, а під *криптологією* — дослідження криптографії та криптології.[2][3]

Дослідження характеристик мов, що мають будь-яке відношення до криптології, таких як частоти появи певних літер, комбінацій літер, загальні шаблони тощо, називається криптолінгвістикою.

## Дія криптографії

Криптографічний алгоритм, або шифр, — це математична формула, що описує процеси шифрування і розшифрування. Щоб зашифрувати відкритий текст, криптоалгоритм працює в сполученні з ключем — словом, числом або фразою. Те саме повідомлення одним алгоритмом, але різними ключами буде перетворюватися в різний шифртекст. Захищеність шифртекста цілком залежить від двох речей: стійкості криптоалгоритму і таємності ключа. Криптоалгоритм плюс усілякі ключі і протоколи, що приводять їх у дію, складають криптосистему. PGP — це криптосистема.

У традиційній криптографії, також називаної шифруванням таємним, або симетричним, той самий ключ використовується як для шифрування, так і для розшифрування даних. Data Encryption Standart (DES) — приклад симетричного алгоритму, що широко застосовувався на Заході з 70-х років у банківській і комерційних сферах. В наш час (2002) його переміняє Advanced Encryption Standard (AES). Малюнок 2 ілюструє процес симетричного шифрування.

## Сучасний стан криптографії

Для сучасної криптографії характерне використання відкритих алгоритмів шифрування, що припускають використання обчислювальних засобів. Відомо більше десятка перевірених алгоритмів шифрування, які, при використанні ключа достатньої довжини і коректної реалізації алгоритму, роблять шифрований текст недоступним для криптоаналізу. Широко використовуються такі алгоритми шифрування як Twofish, IDEA, RC4 та ін.

У багатьох країнах прийняті національні стандарти шифрування. У 2001 році в США прийнятий стандарт симетричного шифрування AES на основі алгоритму Rijndael з довжиною ключа 128, 192 і 256 біт. Алгоритм AES прийшов на зміну колишньому алгоритмові DES, який тепер рекомендовано використовувати тільки в режимі Triple-DES (3DES). В Україні діють наступні стандарти криптографічних перетворень: ДСТУ ГОСТ 28147:2009 — шифрування та формування імітовставки, ДСТУ 4145-2002 —електронний цифровий підпис та ГОСТ 34.311 — геш-функція

# Мережа Фейстеля

**Мережа Фейстеля,** конструкція Фейстеля ( англ. *Feistel network, (Feistel cipher)* - один з методів побудови блокових шифрів . Мережа складається з комірок, званих **осередками Фейстеля.** На вхід кожного осередку надходять дані і ключ. На виході кожного осередку отримують змінені дані і змінений ключ. Всі осередки однотипні, і кажуть, що мережа являє собою певну багаторазово повторювану ( ітерованих ) структуру. Ключ вибирається залежно від алгоритму шифрування / розшифрування і змінюється при переході від одного осередку до іншого. При шифруванні та розшифрування виконуються одні й ті ж операції ; відрізняється тільки порядок ключів . Зважаючи простоти операцій мережу Фейстеля легко реалізувати як програмно, так і апаратно. Більшість сучасних блокових шифрів ( DES , RC2 , RC5 , RC6 , Blowfish , FEAL , CAST-128 , TEA , XTEA , XXTEA та ін.) використовують мережу Фейстеля в якості основи. Альтернативою мережі Фейстеля є підстановлювальний-перестановочне мережу ( AES та ін.).

## Історія

У 1971 Хорст Фейстел ( англ. *Horst Feistel* ) запатентував два пристрої, що реалізують різні алгоритми шифрування , пізніше отримали назву « Люцифер »(« Lucifer »). Один з цих пристроїв використовувало конструкцію, згодом названу «мережею Фейстеля» («Feistel cipher», «Feistel network»). Тоді Фейстел працював над створенням нових криптосистем в стінах IBM разом з Доном Копперсмітом ( англ. *Don Coppersmith* ). Проект «Люцифер» був скоріше експериментальним, але став основою для алгоритму DES ( англ. ***d****ata****e****ncryption****s****tandard).* У 1973 журнал « Scientific American »опублікував статтю Фейстеля «Криптографія і комп'ютерна безпека» («Cryptography and computer privacy») [1] , в якій розкрито деякі важливі аспекти шифрування і наведено опис першої версії проекту « Люцифер ». У першій версії проекту «Люцифер» мережу Фейстеля не використовувалася.

На основі мережі Фейстеля був спроктірован алгоритм DES. У 1977 влади США прийняли стандарт FIPS 46-3, який визнає DES стандартним методом шифрування даних. DES деякий час широко використовувався в криптографічних системах.Ітеративна структура алгоритму дозволяла створювати прості програмні і апаратні реалізації.

Згідно з деякими даними [2] в СРСР вже в 1970-і роки КДБ розробляла блоковий шифр , який використовує мережу Фейстеля, і, ймовірно, саме цей шифр в 1990 був прийнятий як ГОСТ 28147-89 .

У 1987 були розроблені алгоритми FEAL і RC2 . Мережі Фейстеля отримали широке поширення в 1990-і роки - в роки появи таких алгоритмів, як Blowfish (1993), TEA (1994), RC5 (1994), CAST-128 (1996), XTEA (1997), XXTEA (1998), RC6 (1998) та інших.

2 січня 1997 інститут NIST оголосив конкурс зі створення нового алгоритму шифрування даних, покликаного замінити DES . Новий блоковий шифр отримав назву AES ( англ. ***a****dvanced****e****ncryption****s****tandard)* і був затверджений 26 травня 2002 . В AES замість мережі Фейстеля використовується підстановлювальний-перестановочне мережу .

## Конструкція блочного шифру на основі мереж Фейстеля

|  |  |
| --- | --- |
| [https://upload.wikimedia.org/wikipedia/ru/9/97/Feistel_encryption.png](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_encryption.png)  Шифрування | [https://upload.wikimedia.org/wikipedia/ru/4/47/Feistel_decryption.png](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_decryption.png)  Розшифрування |

### Шифрування

Нехай потрібно зашифрувати деяку інформацію , представлену в двійковому вигляді (у вигляді послідовності нулів та одиниць ) і знаходиться в пам'яті комп'ютера або іншого пристрою (наприклад, в файлі ).

Алгоритм шифрування.

* Інформація розбивається на блоки однакової (фіксованого) довжини. Отримані блоки називаються *вхідними,* так як надходять на вхід алгоритму. У випадку, якщо довжина вхідного блоку менше, ніж розмір, який обраний алгоритм шифрування здатний зашифрувати одноразово (розмір блоку), то блок подовжується яким-небудь способом. Як правило довжина блоку є ступенем двійки, наприклад, становить 64 біта або 128 біт.

Далі будемо розглядати операції відбуваються тільки з одним блоком, так як в процесі шифрування з іншими блоками виконуються ті ж самі операції.

* Обраний блок ділиться на два подблока однакового розміру - «лівий» ( L_0 ) І «правий» ( R_0 ).
* «Лівий подблок» L_0 змінюється функцією *f* з використанням раундового ключа K_0 :

x = f (L_0, K_0).

* Результат складається по модулю 2 («xor») з «правим подблоков» R_0 :

x = x \ oplus R_0.

* Результат буде використаний у наступному раунді в ролі «лівого подблока» L_1 :

L_1 = x.

* «Лівий подблок» L_0 поточного раунду буде використаний у наступному раунді в ролі «правого подблока» R_1 :

R_1 = L_0.

* За яким-небудь математичного правилом обчислюється раундовий ключ K_1 - Ключ, який буде використовуватися в наступному раунді.

Перераховані операції виконуються *N-1* раз, де *N* - кількість раундів у вибраному алгоритмі шифрування. При цьому між переходами від одного раунду (етапу) до іншого змінюються ключі: K_0 замінюється на K_1 , K_1 - На K_2 і т.д.).

#### Розшифрування

Розшифровка інформації відбувається так само, як і шифрування, з тим лише виключенням, що ключі слідують у зворотному порядку, тобто не від першого до *N* -ному, а від *N* -го до першого.

### Алгоритмічний опис

* Блок відкритого тексту ділиться на дві рівні частини: (L_0, \ R_0) .
* У кожному раунді обчислюються:

L_i \ = \ R_ {i-1} \ oplus f (L_ {i-1}, K_ {i-1});

R_i \ = \ L_ {i-1},

де:

* *i* - номер раунду ; i = 1 \ ldots N;
* *N* - кількість раундів в обраному алгоритмі шифрування;
* f - Деяка функція;
* K_ {i-1} - ключ *i-1* -го раунду (раундовий ключ).

Результатом виконання N раундів є (L_N, \ R_N) . В *N* -ом раунді перестановка L_N і R_N не проводиться, щоб була можливість використовувати ту ж процедуру і для розшифрування, просто інвертувати порядок використання ключів ( K_N, K_ {N-1}, \ ldots, K_0 замість K_0, K_1, \ ldots, K_N ):

L_ {i-1} \ = \ R_ {i} \ oplus f (L_ {i}, \ K_ {i-1})

R_ {i-1} \ = \ L_i.

Невеликою зміною можна добитися і повної ідентичності процедур шифрування і розшифрування.

Переваги:

* оборотність алгоритму незалежно від використовуваної функції *f;*
* можливість вибору як завгодно складної функції *f.*

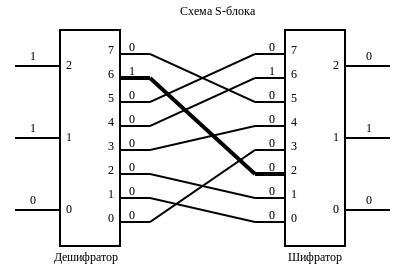
## Функції, використовувані в мережах Фейстеля

У своїй роботі «Криптографія і комп'ютерна безпека» [1] Хорст Фейстел описує два блоки перетворень (функцій f (L_ {i}, \ K_ {i}) ):

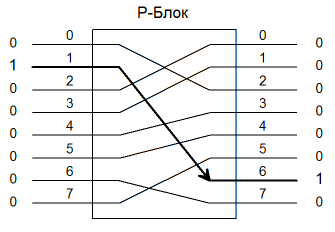
* блок підстановок (s-блок, англ. *s-box);*
* блок перестановок (p-блок, англ. *p-box).*

Можна показати, що будь-яке двійкове перетворення над блоком даних фіксованої довжини, може бути реалізоване у вигляді s-блоку . В силу складності будови *N* -розрядним s-блоку при великих *N* на практиці застосовують більш прості конструкції.

Термін «блок» в оригінальній статті [1] використовується замість терміна «функція» внаслідок того, що мова йде проблоковому шифрі і передбачалося, що s- і p-блоки будуть цифровими мікросхемами (цифровими блоками).

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:S-block.svg)

Принципова схема 3-розрядного s-блоку

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_Network_P-Block.png)

Принципова схема 8-розрядного p-блоку

### S-блок

Блок підстановок (s-блок, англ. *s-box)* складається з наступних частин:

* дешифратор - перетворювач *n* -розряднимдовічного сигналу в однорозрядних сигнал по підставі 2 ^ n ;
* система комутаторів - внутрішні з'єднання (всього можливих з'єднань 2 ^ n! );
* шифратор - перетворювач сигналу з одноразрядного 2 ^ n -річного в n-розрядний двійковий.

Аналіз *n* -розрядним S-блоку при великому n вкрай складний, однак реалізувати такий блок на практиці дуже складно, так як число можливих з'єднань вкрай велике ( 2 ^ n ). На практиці блок підстановок використовується як частина більш складних систем.

У загальному випадку s-блок може мати неспівпадаючі число входів / виходів, в цьому випадку в системі комутації від кожного виходу дешифратора може йти не строго одне з'єднання, а 2 або більше чи не йти зовсім. Те ж саме справедливо і для входів шифратора.

В електроніці можна безпосередньо застосовувати наведену праворуч схему. У програмуванні ж генерують таблиці заміни. Обидва ці підходи є еквівалентними, тобто файл, зашифрований на комп'ютері, можна розшифрувати на електронному пристрої і навпаки.

Таблиця заміни для наведеного 3-розрядногоs-блоку

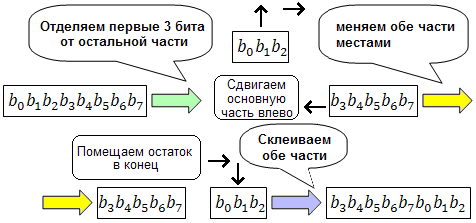
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ комбінації** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **Вхід** | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| **Вихід** | 011 | 000 | 001 | 100 | 110 | 111 | 010 | 101 |

### P-блок

Блок перестановок (p-блок, англ. *p-box)* всього лише змінює положення цифр і є лінійним пристроєм. Цей блок може мати дуже велику кількість входів-виходів, проте в силу лінійності систему не можна вважати кріптоустойчивость.

Криптоаналіз ключа для *n* -розрядним p-блоку проводиться шляхом подачі на вхід *n-1* різних повідомлень, кожне з яких складається з *n-1* нуля («0») і 1 одиниці («1») (або навпаки, з одиниць і нуля ).

### Циклічний зсув

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Left_shift_8_3.png)

Циклічний зсув вліво на 3 розряду 8-бітної шини

Можна показати, що циклічний зсув є окремим випадком p-блоку.

У найпростішому випадку (зсув на 1 біт), крайній біт відщеплюється і переміщується на інший кінець регістру або шини. Залежно від того який біт береться, правий чи лівий, зсув називається вправо або вліво.Зрушення на більше число біт можна розглядати, як багаторазове застосування зсуву на 1.

|  |  |  |
| --- | --- | --- |
| **Циклічний зсув на *m* біт для *n* -розрядним входу *(m <n)*** | | |
| **Напрямок зсуву** | Порядок проходження бітів до зсуву | Порядок проходження бітів після зсуву |
| **Вліво** | b_0, b_1, b_2, ..., b_ {n-1} | b_m, b_ {m + 1}, ... b_ {n-1}, b_0, b_1, ..., b_ {m-1} |
| **Вправо** | b_0, b_1, b_2, ..., b_ {n-1} | b_ {nm}, b_ {nm + 1}, ... b_ {n-1}, b_0, b_1, ..., b_ {nm-1} |

### Додавання по модулю n

Операція « додавання по модулю *n* »позначається як

*(A* + *B)* mod *n*

і являє собою залишок від ділення суми *A* + *B* на *n,* де *A* і *B* - числа.

Можна показати, що складання двох чисел за модулем *n* представляється в двійковій системі числення у вигляді s-блоку, у якого на вхід подається число *A,* а в якості системи комутації s-блоку використовується циклічний зсув вліво на *B* розрядів.

У комп'ютерній техніці і електроніці операція додавання, як правило, реалізована як додавання по модулю n = 2 ^ m , Де *m* - ціле (зазвичай *m* одно розрядності машини). Для отримання в двійковій системі

*A* + *B* mod 2 ^ m

досить скласти числа, після чого відкинути розряди починаючи з *m-того* і старше.

#### Множення по модулю n

Множення по модулю *n* позначається як

*(A* \* *B)* mod *n*

і являє собою залишок від ділення твори *A* \* *B* на *n,* де *A* і *B* - числа.

У персональних комп'ютерах на платформі x86 при перемножуванні двох *m* -розрядних чисел виходить число розрядністю *2 \* m.* Щоб отримати залишок від ділення на 2 ^ m потрібно відкинути *m* старших біт.

### Переваги і недоліки

Переваги:

* простота апаратної реалізації на сучасній електронній базі;
* простота програмної реалізації в силу того, що значна частина функцій підтримується на апаратному рівні в сучасних комп'ютерах (наприклад, додавання по модулю 2 («xor») , додавання по модулю 2 ^ n , Множення по модулю 2 ^ n , І т.д.);
* хороша вивченість алгоритмів, побудованих на основі мереж Фейстеля [4] .

Недоліки:

* за один раунд шифрується тільки половина вхідного блоку [5] .

## Теоретичні дослідження

Мережі Фейстеля були широко вивчені криптографами в силу їх обширного поширення. У 1988 Майкл Люби ( англ.*Michael Luby* ) і Чарльз Ракофф ( англ. *Charles Rackoff* ) провели дослідження мережі Фейстеля і довели, що якщо раундова функція є крипостійкість псевдослучайной, і використовувані ключі незалежні у кожному раунді, то 3-х раундів буде достатньо для того, чтобы блочный шифр являлся псевдослучайной перестановкой, тогда как четырёх раундов будет достаточно для того чтобы сделать сильную псевдослучайную перестановку.

« *псевдовипадковий перестановкою* »Люби і Ракофф назвали таку, яка стійка до атаки з адаптивним вибором відкритого тексту, а« *сильної псевдослучайной перестановкою* »- псевдовипадкову перестановку, стійку до атаки з використанням обраного шифрованого тексту.

Іноді в західній літературі мережу Фейстеля називають «Luby-Rackoff block cipher» на честь Люби і Ракофф, які виконали великий обсяг теоретичних досліджень у цій області.

Надалі, в 1997 році, Моні Наор ( англ. *Moni Naor* ) і Омер Рейнголд ( англ. *Omer Reingold* ) запропонували спрощений варіант конструкції Люби - Ракофф, що складається з чотирьох раундів. У цьому варіанті в якості першого і останнього раунду використовуються дві попарно-незалежні перестановки . Два середніх раунду конструкції Наор - Рейнголда ідентичні раундів в конструкції Люби - Ракофф [6] .

Більшість же досліджень присвячено вивченню конкретних алгоритмів. У багатьох блокових шифрах на основі мережі Фейстеля були знайдені ті чи інші уразливості, проте в ряді випадків ці уразливості є чисто теоретичними і за нинішньої продуктивності комп'ютерів використовувати їх на практиці для злому неможливо.

## Модифікації мережі Фейстеля

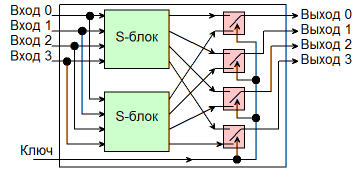
При великому розмірі блоків шифрування (128 біт і більше) реалізація такої мережі Фейстеля на 32-розрядних архітектурах може викликати труднощі, тому застосовуються модифіковані варіанти цієї конструкції. Зазвичай використовуються мережі з 4-я гілками. На малюнку показані найбільш поширені модифікації. Також існують схеми, в яких довжини половинокL_0 і R_0не збігаються. Такі мережі називаються *незбалансованими* .

**Модифікації мережі Фейстеля**

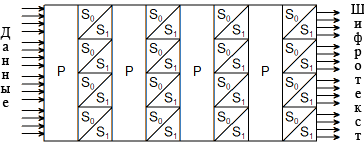
|  |  |  |
| --- | --- | --- |
| [https://upload.wikimedia.org/wikipedia/ru/thumb/3/37/Feistel_type1.png/120px-Feistel_type1.png](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_type1.png)  Тип 1 | [https://upload.wikimedia.org/wikipedia/ru/thumb/8/82/Feistel_type2.png/120px-Feistel_type2.png](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_type2.png)  Тип 2 | [https://upload.wikimedia.org/wikipedia/ru/thumb/d/d4/Feistel_type3.png/120px-Feistel_type3.png](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Feistel_type3.png)  Тип 3 |

## Шифри на основі мережі Фейстеля

### Люцифер (Lucifer)

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:S-chooser.png)

Модуль, що вибирає використовувану таблицю підстановок по битовому ключу

[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Lucifer_v1.png)

Спрощена схема s- і p-шарів в алгоритмі «Люцифер» (червень 1971)

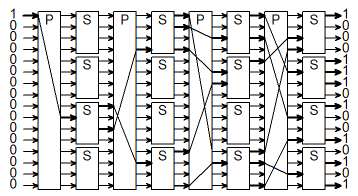
[](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Ones_Propagation.png)

Схема генерації та розповсюдження одиниць

Історично, першим алгоритмом, що використовує мережу Фейстеля, був алгоритм « Люцифер », при роботі над яким Фейстел і була, власне, розроблена структура, що згодом отримала назву «мережа Фейстеля». У червні 1971 Фейстел був отриманий американський патент № 3798359 [8] .

Перша версія « Люцифера »використовувала блоки і ключі довжиною по 48 біт і не використовувала конструкцію« мережа Фейстеля ». Подальша модифікація алгоритму була запатентована на ім'я Джона Л. Смитта ( англ. *John Lynn Smith* ) в листопаді 1971 (US Patent 3,796,830; Nov 1971) [9] , і в патенті міститься як опис власне самої «мережі Фейстеля», так і конкретної функції шифрування. У ній використовувалися 64-розрядні ключі і 32-бітові блоки. І, нарешті, остання версія запропонована в 1973 році і оперувала з 128-бітними блоками і ключами. Найбільш повний опис алгоритму «Люцифер» було наведено в статті Артура Соркіна ( англ. *Arthur Sorkin* ) «Люцифер.Криптографічний алгоритм »(« Lucifer, A Cryptographic Algorithm ») в журналі« Криптологія »(« Cryptologia ») за січень 1984 [10] .

Хоча початкова модифікація «Люцифера» обходилася без «вічок Фейстеля», вона добре демонструє те, як тільки застосуванням s- і p-блоків можна сильно спотворити вихідний текст. Структура алгоритму «Люцифер» зразка червня 1971 являє собою «сендвіч» з шарів двох типів, що використовуються по черзі - так званіSP-мережі (або підстановлювальний-перестановочне мережі). Перший тип шару - p-блок розрядності 128 біт, за ним йде другий шар, що представляє собою 32 модуля, кожен з яких складається їх двох четирёхбітних s-блоків , чиї відповідні входи закорочені і на них подається одне і те ж значення з виходу попереднього шару . Але самі блоки підстановок різні (відрізняються таблицями замін). На вихід модуля подаються значення тільки з одного з s-блоків, якого конкретно - визначається одним з бітів в ключі, номер якого відповідав номеру s-блоку в структурі. Спрощена схема алгоритму меншою розрядності і неповним числом раундів приведена на малюнку. У ній використовується 16 модулів вибору s-блоків (всього 32 s-блоку), таким чином така схема використовує 16-бітний ключ.

Розглянемо тепер, як буде змінюватися шифротекст , в наведеному вище алгоритмі, при зміні всього одного біта.Для простоти візьмемо таблиці замін s-блоків такими, що якщо на вхід s-блоку подаються всі нулі, то й на виході будуть всі нулі. В силу нашого вибору s-блоків, якщо на вхід шифрувального пристрою подаються всі нулі, то й на виході пристрою будуть всі нулі. У реальних системах такі таблиці замін не використовуються, оскільки вони сильно спрощують роботу криптоаналитика, але в нашому прикладі вони наочно ілюструють сильну міжсимвольного взаємозв'язок при зміні одного біта шифруемого повідомлення. Видно, що завдяки першому p-блоку єдина одиниця зсувається переміщається в центр блоку, потім наступний нелінійний s-блок «розмножує» її, і вже дві одиниці за рахунок наступного p-блоку змінюють своє положення і т. Д. В кінці пристрою шифрування, завдяки сильній межсимвольной зв'язку, вихідні біти стали складною функцією від вхідних і від використовуваного ключа. В середньому на виході половина біт буде дорівнює «0» і половина - «1».

За своєю суттю мережу Фейстеля є альтернативою складним SP-мереж і використовується набагато ширше. З теоретичної точки зору раундова функція шифрування може бути зведена до SP-мережі, однак мережа Фейстеля є більш практичною, так як шифрування і дешифрування може вестися одним і тим же пристроєм, але зі зворотним порядком використовуваних ключів. Друга і третя версія алгоритму (що використовують мережу Фейстеля) оперували над 32-бітними блоками з 64-бітовим ключем і 128-бітними блоками з 128-бітними ключами. В останній (третій) версії раундова функція шифрування була влаштована дуже просто - спочатку шіфруемий подблок пропускався через шар 4-бітних s-блоків (аналогічно верствам в SP-мережах, тільки s-блок є константним і не залежить від ключа), потім до нього по модулю 2 додавався раундовий ключ, після чого результат пропускався через p-блок.

### DES

Функція f (L_i, K_i)(де:

* L_i- 32-розрядний вхідний блок на *i* -й ітерації;
* K_i - 48-ми розрядний ключ на даній ітерації)

в алгоритмі DES складається з наступних операцій:

* розширення вхідного блоку *L* до 48-і розрядів (деякі вхідні розряди можуть повторюватися);
* cложеніе по модулю 2 з ключемK_i :

\ Tilde {L_i} = L_i \ oplus K_i;

* поділ результату на 8 блоків довжиною по 6 біт кожен:

\ Tilde {L_i} = \ {\ tilde {L_i} ^ {(0)} \ tilde {L_i} ^ {(1)} \ tilde {L_i} ^ {(2)} \ tilde {L_i} ^ {(3 )} \ tilde {L_i} ^ {(4)} \ tilde {L_i} ^ {(5)} \ tilde {L_i} ^ {(6)} \ tilde {L_i} ^ {(7)} \};

* отримані блоки інформації \ Tilde {L_i ^ {(j)}}подаються на блоки підстановок , що мають 6-ти розрядні входи і 4-розрядні виходи;
* на виході 4-х бітні блоки об'єднуються в 32-х бітний, який і є результатом функції f (L_i, K_i) .

Повне число раундів в алгоритмі DES - 16.

### ГОСТ 28147-89

Функція f (L_i, K_i)(деL_i і K_i - 32-бітові числа) обчислюється таким чином:

* складаються L_i і K_i по модулю 2 ^ {32} :

\ Tilde {L_i} = (L_i + K_i) \ \ bmod \ 2 ^ {32};

* результат розбивається на 8 4-бітових блоків, які подаються на вхід 4-розрядних s-блоків (які можуть бути різними);
* виходи s-блоків об'єднують в 32-бітове число, яке потім зсувається циклічно на 11 бітів вліво;
* отриманий результат є виходом функції.

Кількість раундів в алгоритмі ГОСТ 28147-89 дорівнює 32.

# Література

* *Вербіцький О. В.* Вступ до криптології. — Л.: ВНТЛ, 1998. — 248 с.
* *Бауэр Ф.* Расшифрованные секреты. — М.: Мир, 2007. — 550 с.
* *Брассар Ж.* Современная криптология. — М.: Полимед, 1999. — 176 с.
* *Земор Ж.* Курс криптографии. — Ижевск: РХД, 2006. — 256 с.
* *Мао В.* Современная криптография. — М.: Диалектика, 2005. — 768 с.
* *ван Тилборг Х. К. А.* Основы криптологии. — М.: Мир, 2006. — 472 с.
* *Фергюсон Н., Шнайер Б.* Практическая криптография. — М.: Диалектика, 2004. — 431 с.
* *Шнайер Б.* Прикладная криптография. — М.: Триумф, 2002. — 816 с.
* *Шнайер Б.* Секреты и ложь. — СПб.: Питер, 2003. — 368 с.
* *Ященко В. В.* Введение в криптографию. — М.: МЦНМО, 2012. — 352 с.
* *Oleshchuk V. A.* On Public-Key Cryptosystem Based on Church-Rosser String-Rewriting Systems // Computing and Combinatorics: First Annual International Conference, COCOON '95. — Springer, 1995. — С. 264–269.