Тема 9. СТАНДАРТ ШИФРУВАННЯ ДАНИХ DES, GOST (ГОСТ 28147-89)

На сьогодні розроблено багато симетричних криптосистем, багато з яких є національними стандартами. Найбільш відомі DES, GOST та деякі інші.

Стандарт DES (*Data Encryption Standard*) вперше був опублікований в США у 1975 р. Він був розроблений в корпорації IBM шляхом модифікації більш ранішої версії, що називається LUCIFER. Пізніше з'явилися подальші його модифікації.

ГОСТ 28147-89 – стандарт симетричного шифрування, блоковий алгоритм. Повна назва − «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Алгоритм, що лежить в основі ГОСТа був створений у 8-му головному управлінні КГБ (зараз структура ФСБ), ймовірно ще в 1970 рр. в рамках проектів створення програмних та апаратних реалізацій шифру для різних комп'ютерних платформ. ГОСТ спроектований на забезпечення військового рівня безпеки на 200 років вперед. В 1989 р. він був стандартизований і вперше став офіційним стандартом захисту конфіденційної інформації (специфікація шифру залишалася секретною). В 1994 р. стандарт був розсекречений, опублікований та переведений на англійську мову.

У 2010 році ГОСТ пропонують долучити до ISO 18033 у якості всесвітнього стандарту шифрування. Дуже мала кількість алгоритмів змогли стати міжнародними стандартами. ГОСТ дозволений для захисту секретної інформації без обмежень, у відповідності з тим.

1. Опис стандарту DES

В криптосистемі DES використовується блоковий принцип шифрування двійкового тексту. Довжина блоку шифрування складає 64 біт. Розмір ключа також 64 біт. При цьому кожен восьмий біт є службовим і в шифруванні участі не буре. Кожен такий біт є двійковою сумою семи попередніх і служить лише для знаходження помилок при передачі ключа по каналу зв'язку.

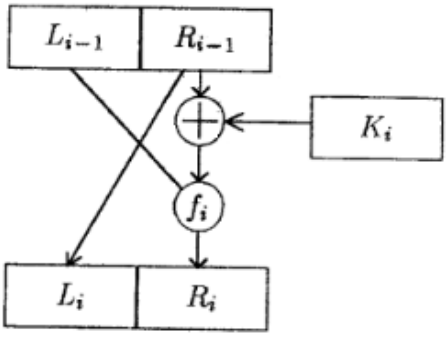
Процес крипто перетворення включає наступні три основні етапи:

1. Біти вхідного повідомлення x переставляються згідно початкової підстановки IP у відповідності таблиці:

IP	58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
	57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Це означає, що 58-й біт стає першим, 50-й – другим і т.д. Далі отриманий вектор $x_0 = IP(x)$ подається у вигляді $x_0 = L_0R_0$, де L_0 – ліва половина із 32 біт, а R_0 – права половина із 32 біт.

2. Повідомлення L_0R_0 перетворюється далі 16 разів по так званій схемі Фейстеля, показаній на мал.1.:



Мал. 9.1 Криптоперетворення Фейстеля

$L_0 = R_{i-1}, R_i = L_{i-1} \oplus f(R_{i-1}, K_i),\ i = 1,2,\dots 16,$ де функція f і опис ключів $K_1, K_2, \dots, K_{16}$, будуть описані детально.

3. Повідомлення $L_{16}R_{16}$ перемішується підстановкою IP^{-1}: $y = IP^{-1}(L_{16}R_{16})$ є зашифроване повідомлення.

Шифрування відбувається за схемою, наведеною на мал. 9.2. [13]

2. Функція f

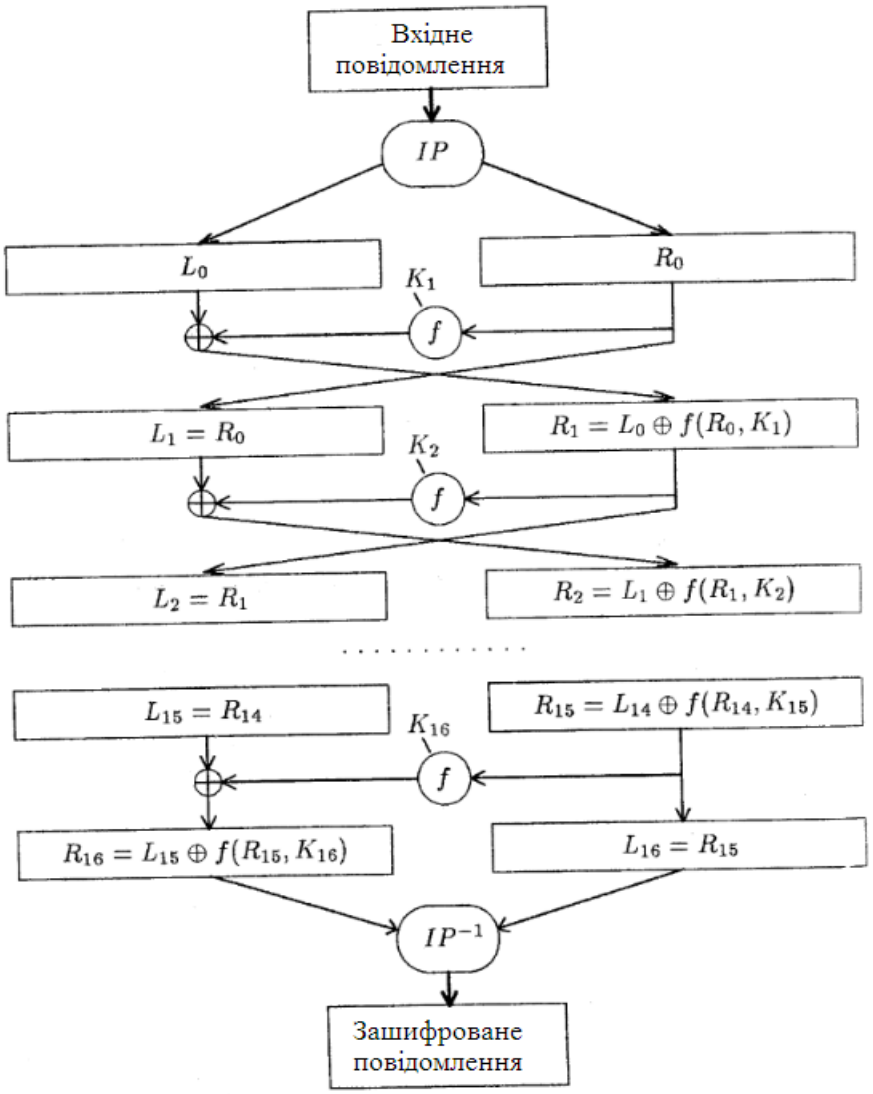
Ця функція має два аргументи *A, B*. Перший з них складається з 32 біт, а другий із 48 біт. Результат має 32 біт.

1. Перший аргумент *A*, що має 32 біт, перетворюється в 48-бітовий вектор *P(A)* шляхом перестановки з повторенням початкового вектора *A*. Ця процедура одна і та ж для всіх раундів. Вона задається таблицею:

P_1	32	1	2	3	4	5	4	5	6	7	8	9	8	9	10	11
	12	13	12	13	14	15	16	17	16	17	18	19	20	21	20	21
	22	23	24	25	24	25	26	27	28	29	28	29	30	31	30	1

2. Далі обчислюється сума $P(A) \oplus B$ і записується у вигляді конкатенації восьми 6-бітових слів: $P(A) \oplus B = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$.

3. На цьому етапі кожне слово B_i поступає на відповідний блок *S*-блок S_i. Блок S_i перетворює 6-бітовий вхід B_i в 4-бітовий вихід C_i. *S*-блок – це матриця 4 х 16 з цілими елементами в діапазоні від 0 до 16. Два перших біти слова B_i, якщо їх розглядати як двійковий запис числа, визначають номер рядка матриці *S*-блока. Чотири останніх біти визначають деякий стовпець. Тим самим знайдений деякий елемент матриці. Його двійковий запис і є виходом. [13]



Мал. 9.2 Схема криптоперетворення DES

В таблицях представлені всі 8 блоків DES.

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2	13	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

4. Вихід $C = C_1 C_2 \ldots C_S$ перемішується фіксованою підстановкою P_2:

P_2																
	16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
	2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

3. Розписування ключів

1. В 64-бітовому ключі K видаляються біти 8, 16, …, 64. Біти, що лишилися, перемішуються підстановкою P_3:

P_3														
	57	49	41	33	25	17	9	1	58	50	42	34	26	18
	10	2	59	51	43	35	27	19	11	3	60	52	44	36
	53	55	47	39	31	23	15	7	62	54	46	38	30	22
	14	6	61	53	45	37	29	21	13	5	28	20	12	4

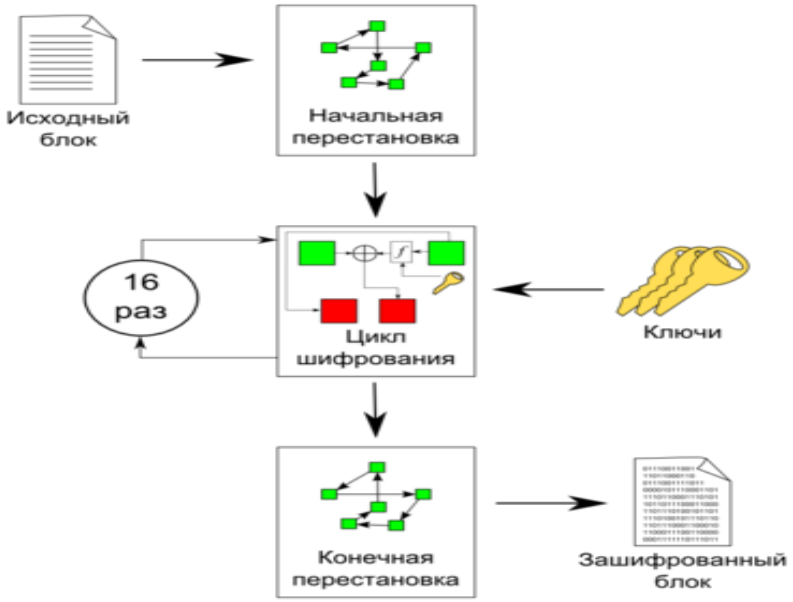
Вихід $P_3(K)$ подається у вигляді $P_3(K) = C_0 D_0$, де C_0 – ліва половина, D_0 – права половина.

2. Чергові C_i, D_i обчислюються за схемою $C_i = L_i(C_{i-1})$, $D_i = L_i(D_{i-1})$, де L_i – циклічний зсув вліво на одну позицію, якщо *i=1, 2, 3, 9, 16*. В решту випадках L_i – зсув вліво на дві позиції. [13]

3. На цьому етапі вихід перемішується підстановкою P_4:

P_4												
	14	17	11	24	1	5	3	28	15	6	21	10
	23	19	12	4	26	8	16	7	27	20	13	2
	41	52	31	37	47	55	30	40	51	45	33	48
	44	49	39	56	34	53	46	42	50	36	29	39

Схематично робота алгоритму виглядає так:

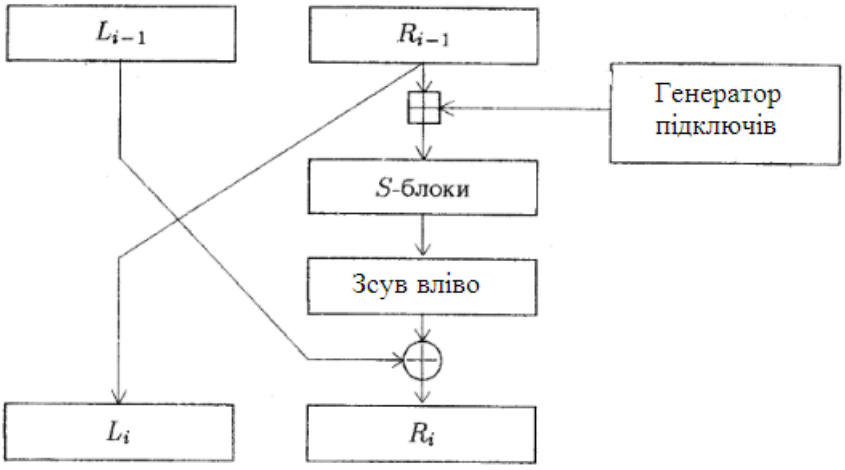


Дешифрування здійснюється тим самим алгоритмом і ключем, але в розписування ключів вносяться деякі зміни: міняють на зворотній порядок генерування ключів. Іншими словами, дешифрування відбувається зворотнім способом.

З ціллю збільшення стійкості алгоритму DES було розроблено декілька його модифікацій: double DES (2DES), triple DES (3DES), DESX, G-DES.

4. Вітчизняний аналог DES – специфікований ГОСТ 28147-89

ГОСТ є 64-бітовим 32-раундовим алгоритмом з 256-бітовим ключем. *S*-блоки, які є в алгоритмі, також можна використовувати як ключ. При шифруванні повідомлення представляється у вигляді конкатенації двох половинок *LR*. Один раунд алгоритму здійснює своє перетворення Фейстеля, мал. 3.: $L_i = R_{i-1}, R_i = L_{i-1} \oplus f\ (R_{i-1}, K_i)$.



Мал. 9.3 Один раунд ГОСТа

Відмінність від стандарту DES – лише в конструкції функції *f*. Робота алгоритму закінчується транспозицією: $LR \rightarrow RL$.

Опис функції f

Спочатку права половина R_i додається по модулю 2^{32} з підключем K_i.

Отримане 32-бітове повідомлення ділиться на вісім 4-бітових частин. Кожна з цих 4-бітових чисел перетворюється відповідним *S*-блоком в інше 4-бітове число.

Будь який *S*-блок визначається деякою 16-бітовою перестановкою на множині 16 елементів 0, 1, 2, … 15. В якості *S*-блоків можуть використовуватися різні таблиці чисел. Наприклад, в одному з банків використовуються *S*-блоки:

S_1 = (4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3),

S_2 = (14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9),

S_3 = (5 8 1 13 10 9 4 2 14 15 12 7 6 0 9 11),

S_4 = (7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3),

S_5 = (6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2),

S_6 = (4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14),

S_7 = (13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12),

S_8 = (1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12).

Після перетворення *S*-блоками отримане 32-бітове повідомлення зсувається вліво на 11 позицій.

Ключ

Початковий 256-бітовий ключ ділиться на вісім 32-бітових підключів k_1, k_2, …, k_8; вони використовуються в 32 раундах в наступному порядку: 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1.

При дешифруванні порядок використання підключів міняється на протилежний. [13]

Є дві основні модифікації алгоритму ГОСТ: GOST-H и GOSTA. За результатами аналізу був зроблений висновок про те, що GOST-H і GOSTA слабші від початкового алгоритму ГОСТ 28147-89, оскільки обидва мають класи слабких ключів.

За словами основоположника алгебраїчного криптоаналізу Ніколая Куртуа блоковий шифр ГОСТ, який найближчим часом повинен був стати міжнародним стандартом, фактично взломаний, іншими словами, знайдені дуже важливі слабкі місця. (травень 2011)

5. Порівняння шифрів ГОСТ 28147-89 і DES

Порівняння основних характеристик цих двох алгоритмів наведені в таблиці:

Параметр	ГОСТ	DES
Розмір блоку шифрування	64 біт	64 біт
Довжина ключа	256 біт	56 біт
Число раундів	32	16
Вузли заміни (S-блоки)	не фіксовані	фіксовані
Довжина ключа для одного раунду	32 біт	48 біб
Схема вироблення раундового ключа	проста	складна
Початкова та кінцева перестановка бітів	немає	є

Функція шифрування ГОСТа набагато простіша функції шифрування DES, вона не містить операцій бітових перестановок, які неефективно реалізуються на сучасних універсальних процесорах. Але не дивлячись на те, що у ГОСТа удвічі більше раундів (32 проти 16), програмна реалізація на процесорах Intel x86 більш ніж у 2 рази краща по швидкодії від реалізації DES.

На кожному раунді шифрування використовується “раундовий ключ”, в DES він 48-бітовий і виробляється за відносно слабким алгоритмом, що включає бітові перестановки і заміни за таблицею, в ГОСТі він береться як фрагмент ключа шифрування. Довжина ключа шифрування в ГОСТі 256 біт, довжина раундового – 32 біт, отримуємо, що ключ шифрування містить 256/32=8 раундових ключів. В ГОСТі 32 раунди, відповідно, кожний раундовий ключ використовується 4 рази, порядок використання раундових ключів встановлений і різний для різних режимів.

Таблиця замін в ГОСТі – аналог S-блоків DES – являє собою таблицю (матрицю) розміром 8x16, що містить числа від 0 до 15. В кожному рядку кожне з 16 чисел повинно зустрітися тільки один раз. На відміну від DES, таблиця замін в ГОСТі одна і та ж для всіх раундів і не зафіксована в стандарті, а є змінним секретним ключовим елементом.

В ГОСТі на відміну від DES, немає початкової і кінцевої бітових перестановок шифрованого блоку, які, за думкою багатьох спеціалістів, не впливають суттєво на стійкість шифру, хоча впливають (в сторону зменшення) на ефективність його реалізації.