ЛАБОРАТОРНА РОБОТА № 3. МОДЕЛЮВАННЯ ПРОЦЕСІВ ШИФРУВАННЯ ЗА ДОПОМОГОЮ ОПЕРАЦІЇ XOR. АЛГОРИТМ DES

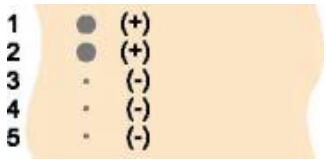
Мета роботи: набути вміння шифрування повідомлень із використанням операції побітового додавання за модулем 2, дослідити процеси шифрування за допомогою алгоритму DES на основі навчальної програми CrypTool 2.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням MS Excel та CrypTool 2, текстові повідомлення згідно варіанту.

Теоретичні відомості

ШИФР ОДНОРАЗОВОГО БЛОКНОТУ (ШИФР ВЕРНАМА)

Шифр одноразового блокноту, або шифр Вернама, було запропоновано у 1917 році співробітниками телеграфної компанії AT&T *Мейджором Джозефом Моборном* та *Гільбертом Вернамом*. Відкритий текст представлявся у вигляді п’ятизначних імпульсних комбінацій – коді Бодо. Наприклад, літера «А» на паперовій стрічці мала вигляд (рис. 3.1):



«+» – отвір;
«-» – його відсутність.

Рис. 3.1. Літера «А» на паперовій стрічці

У класичному розумінні одноразовий блокнот є унікальною послідовністю символів ключа, що згенерована випадковим чином. Заздалегідь готувалася «гама» – перфострічка з випадковими знаками. Потім електромеханічно складалися її імпульси з імпульсами знаків відкритого тексту. Отримана сума представляла собою шифротекст. На приймальному кінці імпульси, отримані по каналу зв'язку, складалися з імпульсами тієї ж самої «гами», в результаті чого відновлювалися вихідні імпульси повідомлення.

Ідея шифру Вернама легко поширюється на двійкові дані. Ключем виступає послідовність випадкових символів. При цьому ключ повинен володіти трьома критично важливими властивостями:

1) бути дійсно випадковим;

2) за розміром збігатися з заданим відкритим текстом (ключ ні в якому разі не зациклюється);

3) застосовуватися тільки один раз.

Для шифрування бінарних даних (потоку бітів) виконується додавання бітів за модулем 2 (операція XOR, eXclusive OR – виключне або), що позначається ⊕ (табл. 3.1).

Таблиця. 3.1. Операція XOR над бітами

⊕	**0**	**1**
0	0	1
1	1	0

На практиці використовують довгі випадкові або псевдовипадкові ключі, згенеровані за допомогою спеціальних технічних пристроїв або програмно-апаратних комплексів. Можна один раз фізично передати носій інформації з довгим дійсно випадковим ключем, а потім по мірі необхідності пересилати повідомлення. При дешифруванні одержувач, використовуючи точно такий самий ключ, виконує додавання за модулем 2 кожного символу ключа та шифротексту.

Приклад 3.1:

Шифрування за допомогою шифру одноразового блокнота повідомлення *SUN* із використанням випадкової ключової послідовності 00001011 00010010 00001111:

Відкритий текст	01010011 01010101 01001110
Ключова гама	00001011 00010010 00001111
Результат додавання за модулем 2	01011000 01000111 01000001
Шифротекст	*XGA*

У 1949 році Клод Шеннон опублікував роботу, в якій довів абсолютну стійкість шифру Вернама. Інших шифрів з цією властивістю не існує. При цьому умови, яким повинен задовольняти ключ, настільки сильні, що практичне використання шифру Вернама є важко здійсненним. Тому він використовується тільки для передачі повідомлень найвищої секретності.

АЛГОРИТМ DES

Американський стандарт шифрування даних (Data Encryption Standard), оснований на мережі Фейстеля та прийнятий у 1977 році, є типовим представником сімейства блокових шифрів.

Ключ шифрування складається з 56 випадкових бітів; додається ще 8 біт в позиціях 8, 16,…, 64, таким чином, щоб кожен байт містив непарну кількість одиниць. (використовується при знаходженні помилок при обміні та зберіганні ключів).

Процес шифрування полягає в початковій перестановці 64 бітів вхідного блоку, шістнадцяти циклах шифрування та кінцевій перестановці бітів (рис. 3.2).

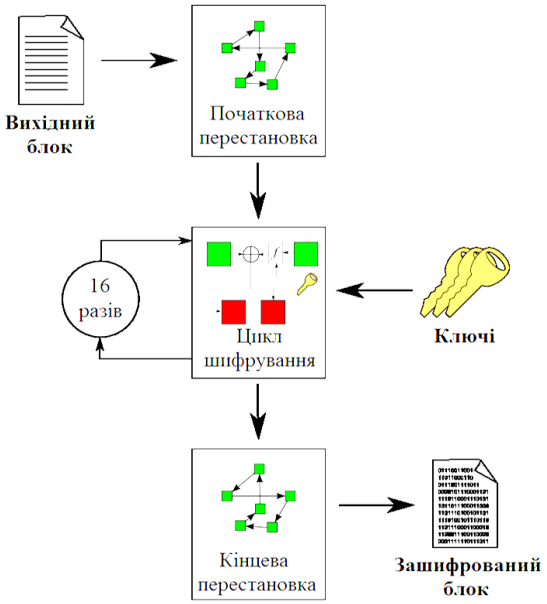


Рис. 3.2. Загальна схема алгоритму DES

Розглянемо алгоритм докладніше:

Початкова перестановка

Початковий текст, що являє собою 64-бітний блок $X = (x_1, x_2, x_3, \ldots, x_{64})$ перетворюється в 64-бітний блок $X_0 = IP(X)$ за допомогою початкової перестановки *IP* (Initial Permutation), що визначається таблицею 3.2.

Таблиця 3.2. Матриця початкової перестановки 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

Раунди шифрування

Після *IP*-перестановки 16 разів повторюється процедура шифрування блоку X_0 за допомогою функції f та раундових ключів K_i, де $i = 1, 2, \ldots, 16$ (рис. 3.3).

Кожен раунд шифрування містить такі етапи:

1. $X_0 = IP(X)$ розбивається на дві половини L_0, R_0, *де* L_0 – перші (старші) 32 біти блоку X_0, а R_0 – останні (молодші) 32 біти блоку X_0.
2. Права половина R_i – це бітове додавання L_{i-1} та $f(R_{i-1}, K_i)$ по модулю 2: $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$.
3. Ліва половина L_i дорівнює правій половині попереднього блоку R_{i-1} без змін: $L_i = R_{i-1}$.

Після 16-ї ітерації ліва і права половини блока не міняються місцями.

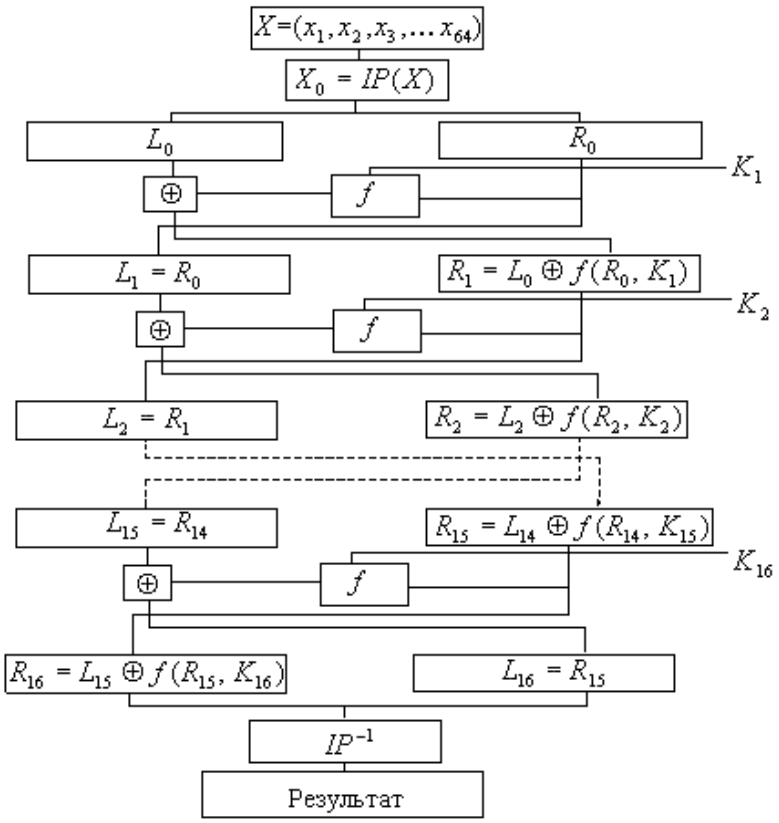


Рис. 3.3. Схема шифрування алгоритму DES

Основна функція шифрування (функція Фейстеля)

Аргументи функції f – 32-бітовий вектор R_{i-1} та 48-бітовий підключ K_i. Для обчислення функції f використовуються:

1) функція розширення E;
2) перетворення S, яке складається з 8 перетворень *S*-блоків;
3) перестановка P.

Функція E розширює 32-бітовий вектор R_{i-1} до 48-бітового вектора $E(R_{i-1})$ шляхом дублювання деяких бітів R_{i-1}. Порядок бітів вектора $E(R_{i-1})$ зазначений у таблиці 3.3.

Таблиця 3.3. Перестановка з розширенням

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	32	1

Отриманий після розширення блок $E(R_{i-1})$ додається по модулю 2 із раундовими ключами K_i. Потім представляється у вигляді восьми послідовних блоків $B_1, B_2, \ldots, B_8$, тобто $E(R_{i-1}) \oplus K_i = B_1 B_2 \ldots B_8$.

Кожен B_j являється 6-бітовим блоком. Далі кожен з блоків B_j перетворюється у 4-бітовий блок B'_j за допомогою перетворень S_j. Перетворення S_j визначаються таблицею 3.4. Індекс j вказує, який з масивів *S*-боксу використовувати. Застосувавши операцію вибору до кожного із блоків B_j, одержимо 32-бітний блок $B'_1, B'_2, \ldots, B'_8$.

Таблиця 3.4. S-бокси алгоритму DES

		Номер стовпця															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
Номер рядка	**0**	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	**S1**
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
	2	4	1	14	89	13	6	2	11	15	12	9	7	3	10	5	0	
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	**S2**
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	**S3**
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	**S4**
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	**S5**
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	

	Номер стовпця																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	**S6**
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	**S7**
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	**S8**
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Приклад 3.2:

Припустимо, що B_3 = 101111. Знайдемо B_3' – ?

Перший і останній розряди B_3 – двійковий запис числа *a,* 0≤*a*≤3.

Середні чотири розряди B_3 – двійковий запис числа *b*, 0≤*b*≤15.

Пара чисел (*a*,*b*) визначає число, що знаходиться в перетині рядка *a* та стовпця *b*. Двійкове представлення цього числа дає B_3'.

У нашому випадку $a = 11_2 = 3$, $b = 0111_2 = 7$, а число обумовлене парою (3,7), дорівнює 7. Його двійкове представлення B_3' = 0111.

Отриманий блок $B_1', B_2', \ldots, B_8'$ перетворюється за допомогою матриці перестановки P (табл. 3.5).

Таблиця 3.5 Матриця перестановки P

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

Таким чином, $f(R_{i-1}, K_i) = P(B_1', B_2', \ldots, B_8')$.

Генерація ключів

Ключ K – 64-бітний блок з вісьмома бітами контролю парності, що розміщені в позиціях 8, 16, 24, 32, 40, 48, 56, 64. Ще раз відзначимо, що на кожній ітерації використовується нове значення ключа $K_1, K_2, \ldots, K_{16}$, яке обчислюється із початкового значення ключа K.

Для видалення контрольних бітів і підготовки ключа до роботи використовується перестановка ключа (табл. 3.6).

Таблиця 3.6. Матриця перестановки ключа

57	49	41	33	25	17	9	C_0
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	D_0
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

Ця перестановка визначається двома блоками C_0 та D_0 по 28 біт кожний. Тобто 56-бітовий ключ ділиться на 2 половини, які потім циклічно зсуваються на один чи два біти ліворуч в залежності від етапу.

Тобто C_i, D_i, де i = 1,2,3,...,16 визначаються з C_{i-1}, D_{i-1}, одним або двома лівими циклічними зсувами згідно таблиці. 3.7.

Таблиця 3.7. Матриця зсуву для обчислення ключів

Раунд	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число зсуву	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Після зсуву C_iD_i знову вибирається 48 бітів з 56 бітів та міняється їх порядок за наступною таблицею (табл. 3.8):

Таблиця 3.8. Матриця перестановки зі стисненням

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
44	49	39	56	34	53
46	42	50	36	29	32

Наприкінці шифрування виконується відновлення позицій бітів за допомогою матриці перестановок IP^{-1} (табл. 3.9).

Таблиця 3.9. Матриця кінцевої перестановки IP^{-1}

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

При дешифруванні даних всі дії відбуваються в зворотному порядку. Ключі застосовуються в зворотному порядку. Функція *f*, перестановки *IP* і IP^{-1} такі самі як і в процесі шифрування.

ЗНАЙОМСТВО ІЗ СЕРЕДОВИЩЕМ CRYPTOOL 2

CrypTool 2 – безкоштовне програмне забезпечення з відкритим вихідним кодом, що реалізує концепцію візуального програмування та виконання каскадів криптографічних процедур. CrypTool 2 є однією із складових великого проекту CrypTool, призначеного в першу чергу для електронного навчання криптографії та криптоаналізу.

Програмний засіб CrypTool 2 (рис. 3.4) на даний час доступний німецькою та англійською мовами, має інтуїтивно зрозумілий сучасний графічний інтерфейс та зручне меню, за допомогою якого користувач у робочій області програми може перетворювати повідомлення з використанням найвідоміших криптографічних алгоритмів.

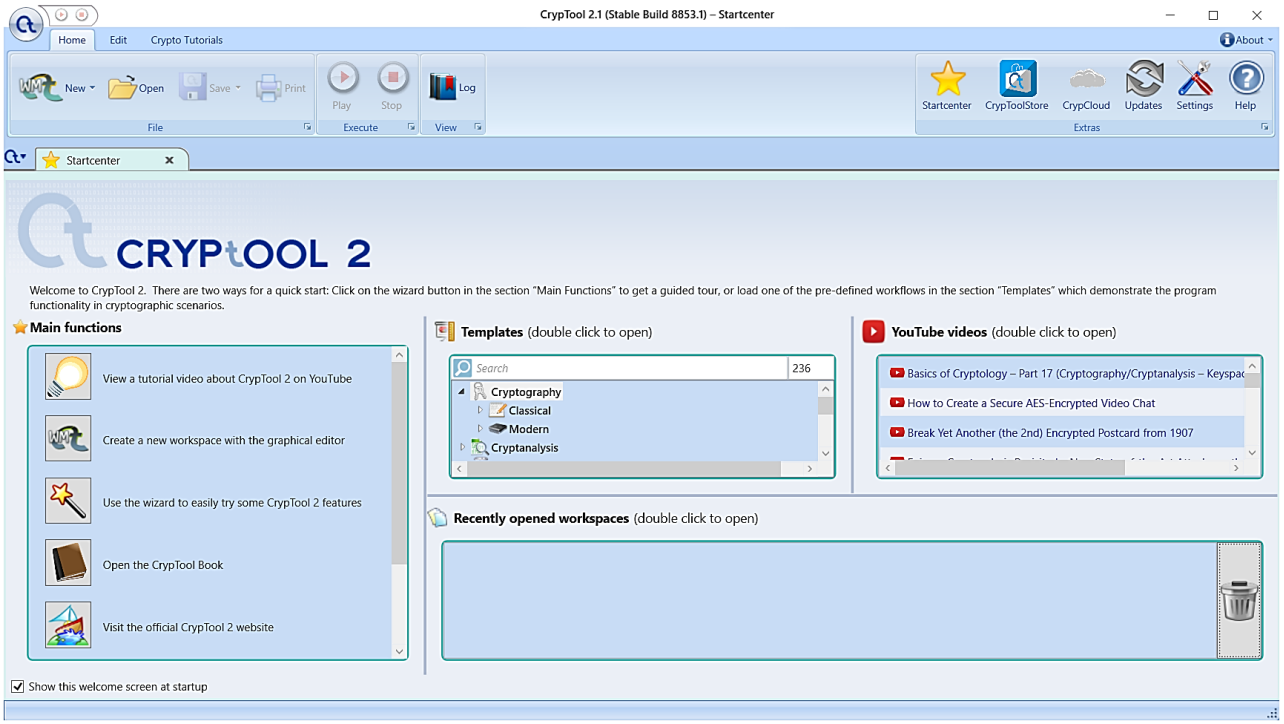


Рис. 3.4. Вікно завантаження CrypTool 2

Розділ «Templates» (рис. 3.5) містить як готові шаблони проектів, що реалізують криптографічні та криптоаналітичні алгоритми, математичні та інші функції, протоколи тощо, так і набір інструментів, котрі можуть бути використані для модифікації готових або створення нових проектів. Крім того CrypTool 2 пропонує

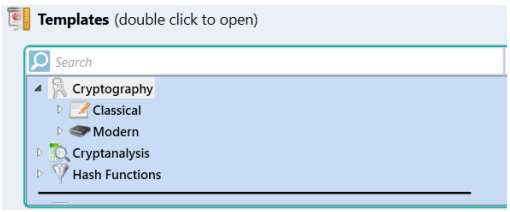


Рис. 3.5. Шаблони CrypTool 2

стеганографічні способи перетворення даних, тобто такі, при яких повідомлення не шифрується, а приховується сам факт його передачі чи існування.

Для використання готового шаблону у розділі «Templates» із меню, необхідно обрати потрібний алгоритм. Після чого відкриється нова вкладка, що складатиметься із окремих модульних компонентів, пов'язаних між собою (рис. 3.6). Вони мають властивості подібні до діалогового вікна операційної системи Windows. Активізація компоненту відбувається шляхом натискання по ньому лівою клавішею миші. Кожен компонент у лівому верхньому кутку містить меню (наприклад, ), що дає змогу налаштування дій, відкриття довідки тощо.

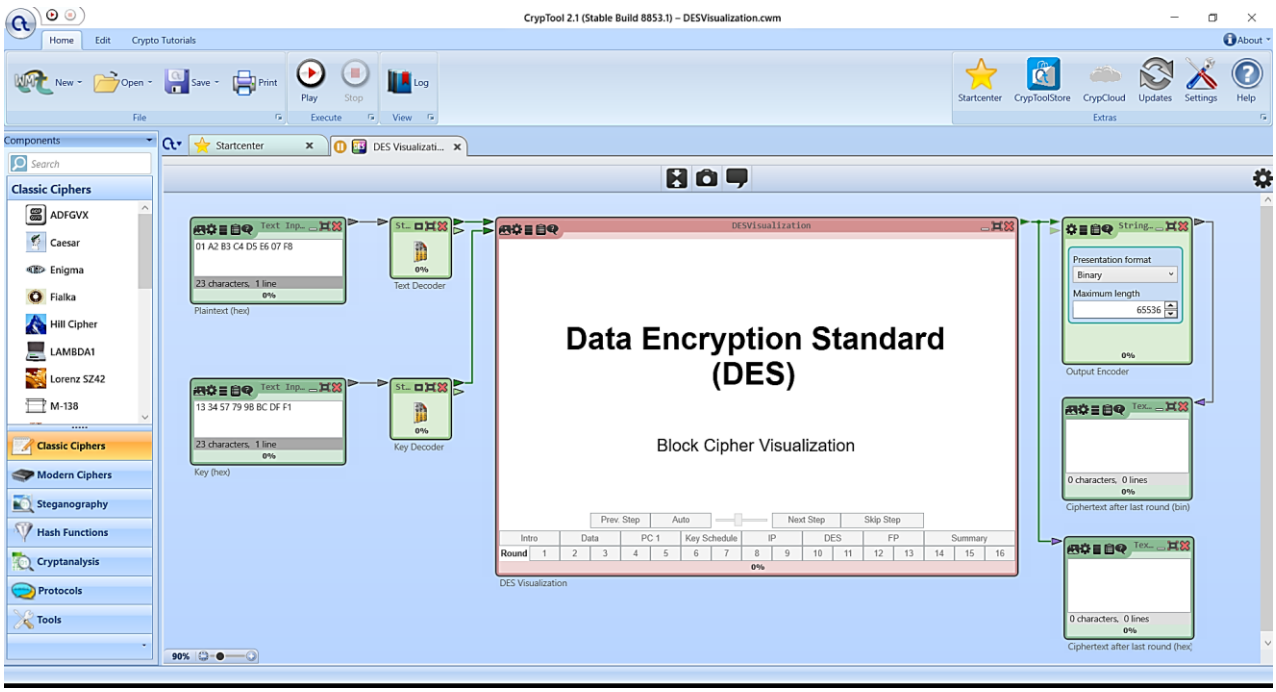


Рис. 3.6. Вкладка шаблону, що візуалізує шифр DES

Налаштування параметрів роботи компонентів (наприклад, визначення дії криптографічного перетворення, алфавіт, ключ тощо) відбувається з використанням панелі «Parameter», що знаходиться праворуч робочого вікна, або за допомогою опції «Settings» у меню компоненту. Панель модульних компонентів «Components», що розміщена ліворуч робочого вікна, дає змогу додавати до проекту нові складові, таким чином удосконалюючи його.

Завдання до лабораторної роботи

Завдання 1

Реалізувати в середовищі MS Excel або на будь-якій мові програмування перетворення текстового повідомлення у двійкову послідовність, що додається

за модулем 2 (XOR) із ключовою гамою. Перевірити роботу програми для вхідних даних згідно варіанту (англійській алфавіт). Кроки алгоритму шифрування зі скріншотами описати у звіті.

Варіант №	Відкритий текст	Ключова гама
1.	JOB	00000010 01111110 00010110
2.	LAW	00000001 01100101 01111000
3.	CAT	00010011 01101001 00011100
4.	AGE	01100000 00000010 00001010
5.	TEA	00011111 00011100 01110101
6.	SKY	01111000 01100100 00011101
7.	WIN	01100000 01101000 00011101
8.	ICE	01111111 00001011 01110010
9.	DAY	01100010 00011000 00100010
10.	ART	00001001 00010110 00011110
11.	JOY	00000011 01100100 01100011
12.	SEA	01111001 00011100 01101010
13.	BAG	01101110 01101110 00011100
14.	JAM	00011101 01100100 00011011
15.	OWL	01111000 00100001 00110010

Завдання 2

Виконати зашифрування блоку повідомлення за допомогою алгоритму DES на основі навчальної програми CrypTool 2 (Templates⇒Cryptography⇒ Modern⇒Symmetric⇒DES Visualization). Ключ ***обрати самостійно*** *(64 бітова послідовність символів), не використовувати ключ за замовчуванням. У звіті описати зі скріншотами нижчезазначені кроки алгоритму.*

Варіант №	Відкритий текст
1.	EVERYONE
2.	TOMORROW
3.	DOWNLOAD
4.	KINDNESS
5.	BIRTHDAY
6.	INFINITY
7.	ORIGINAL
8.	POSITIVE
9.	DAUGHTER
10.	GRATEFUL
11.	PROPERTY
12.	EXCHANGE
13.	CHAMPION
14.	PROGRESS
15.	MAGAZINE

Генерація ключів

- ✓ Ключ (64 біти) у 16-ій системі числення.
- ✓ Ключ (64 біти) у 2-ій системі числення (з позначенням бітів контролю парності).
- ✓ Ключ (56 біт) у 2-ій системі числення.
- ✓ Перестановка початкового 56-бітного ключа.
- ✓ Поділ ключа на дві частини C_0 та D_0.
- ✓ Зсув C_0 та D_0.
- ✓ Перестановка зі стисненням.
- ✓ Таблиця усіх раундових ключів.

Зашифрування блоку

- ✓ Початковий блок повідомлення у вигляді тексту.
- ✓ Початковий блок повідомлення (64 біти) у 16-ій системі числення.
- ✓ Початковий блок повідомлення (64 біти) у 2-ій системі числення.
- ✓ Початкова перестановка.
- ✓ Поділ блоку на дві половини L_0 та R_0 (1 раунд).
- ✓ Функція Фейстеля та додавання з ключем:
  - Перестановка з розширенням R_0;
  - Додавання R_0 з раундовим ключем K_1;
  - Перетворення з використанням S-боксів (з поясненням);
  - Перестановка P.
- ✓ Додавання L_0 та $f(R_0, K_1)$.
- ✓ Таблиця лівих та правих половин кожного раунду.
- ✓ Кінцева перестановка.
- ✓ Результат шифрування блоку.

Контрольні запитання:

1. У чому полягає алгоритм одноразового блокноту?
2. Що являє собою операція XOR?
3. Які переваги і недоліки шифрування методом одноразового блокноту?
4. До яких шифрів належить стандарт шифрування даних DES?
5. Якою повинна бути довжина ключа у шифрі DES?
6. З яких кроків складається алгоритм шифрування DES.
7. Скільки разів виконується перетворення Фейстеля над блоком у DES?