Akademia Techniczno-Humanistyczna
w Bielsku-Białej

Wydział Budowy Maszyn i Informatyki

Laboratorium z sieci komputerowych

Ćwiczenie numer:
11

Temat ćwiczenia:

System szyfrowania danych DES.

1. Wstęp teoretyczny.

DES (Data Encryption Standard) jest jednym z najpopularniejszych algorytmów szyfrowania danych (kryptosystemów). Opracowano go w latach siedemdziesiątych w firmie IBM, w ramach konkursu na stworzenie efektywnego kryptosystemu na potrzeby rządu Stanów Zjednoczonych. Po drobnych modyfikacjach wprowadzonych przez NSA (National Security Agency - Narodową Agencję Bezpieczeństwa), w 1977 roku DES został uznany przez rząd USA za oficjalny standard. Od tej pory jest szeroko wykorzystywany - głównie w świecie finansów i bankowości. Danymi wejściowymi algorytmu DES mogą być: text, plik lub dowolne dane w postaci binarnej.

Zasada działania

DES szyfruje 64-bitowe bloki danych przy użyciu klucza o długości 64 bitów, przy czym informacji użytecznej w kluczu jest 56 bitów, gdyż co ósmy bit jest bitem parzystości. Operacja przebiega w kilku-kilkunastu etapach, podczas których tekst wiadomości ulega wielokrotnym przeobrażeniom. Tak jak w każdej metodzie kryptograficznej posługującej się kluczem prywatnym, klucz ten musi być znany zarówno nadawcy jak i odbiorcy. Ponieważ dla każdej wiadomości klucz wybierany jest losowo spośród 72 000 000 000 000 000 (72 kwadrylionów) możliwych, wiadomości szyfrowane przy pomocy algorytmu DES przez długi czas uchodziły za niemożliwe do złamania. Obecnie długość klucza stanowi największą wadę algorytmu.

Opis algorytmu DES

Schemat blokowy algorytmu DES:

Pierwszym krokiem jest przestawienie 64 bitów bloku wejściowego w sposób określony permutacją początkową (inicjującą) IP. Według niej w miejsce pierwszego bitu wstawiana jest wartość bitu 58, w miejscu drugiego bit 50 itd.

Permutacja IP odbywa się według poniższej tabeli:

Permutacja 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

Kolejnym krokiem jest podział bitów na dwie części lewą – L i prawą – R po 32 bity każda.

Wartości z bloku R_{i-1} (gdzie i jest licznikiem kolejnych cykli) przepisywane są do bloku L_i oraz poddawane funkcji F z parametrem K_i (gdzie K_i to klucz cyklu i). Wartości bloku L_{i-1} wraz z danymi otrzymanymi z funkcji F poddawane są operacji XOR. Wynik operacji przepisywany jest do bloku R_{i+1}.

Operacje te powtarzane są w kolejnych cyklach od 1 do 15. Ostatnim cyklem jest cykl numer 16, przebiega on identycznie jak poprzednie piętnaście cykli z tą jedynie różnicą, iż występuje w nim wymiana zawartości bloków R oraz L.

Po zakończeniu 16 cyklów następuje złączenie obu bloków R i L w całość (64 – bity). Bity te poddaje się permutacji końcowej IP^{-1}.

Permutacja IP^{-1} odbywa się według poniższej tabeli:

Permutacja 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

Dane wyjściowe z permutacji końcowej IP^{-1} kończą proces szyfrowania 64 – bitowego bloku danych.

Schemat blokowy funkcji $\mathbf{F(R_{i-1}, K_i)}$ znajduje się poniżej:

Pierwszym krokiem w funkcji F jest dokonanie ekspansji E. Jest to operacja, dzięki której z 32 bitów bloku R_{i-1} tworzony jest blok o wielkości 48 bitów. Ekspansja dokonywana jest w celu dostosowania długości danych do długości klucza K_i, aby możliwe było wykonanie operacji XOR. Ekspansję E wykonuje się w analogiczny sposób jak permutacje IP oraz IP^{-1}.

Ekspansja E odbywa się według tabeli znajdującej się poniżej:

Ekspansja E					
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

Po dokonaniu ekspansji otrzymane dane poddawane są operacji XOR razem z wartościami klucza K_i .

Otrzymane wartości są następnie rozdzielane na 8 bloków B (każdy po 6 bitów). Bloki te są numerowane kolejno od 1 do 8 (B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8). Bloki te poddawane są operacji podstawienia przy użyciu ośmiu S-bloków (z zasadą blok B_1 wykorzystuje S-blok S_1). Podstawienie to przeprowadza się w następujący sposób. Pierwszy i ostatni bit w bloku B reprezentuje binarną wartość wiersza r w S-bloku. Bity od drugiego do piątego reprezentują binarną wartość kolumny c w S-bloku. Mając już wyznaczony numer wiersza r i kolumny c, odczytujemy konkretną wartość z S-bloku. Jest to zawsze wartość z zakresu 0-15 zatem można ją zapisać w postaci 4 bitów. Po dokonaniu podstawień dla wszystkich ośmiu bloków B otrzymujemy osiem czterobitowych wartości, co razem daje 32 bity.

Przykład:

Mając blok B_1 = 10010010 wyznaczamy wiersz r = 10 (2 – dziesiętnie) oraz kolumnę c = 001001 (9 – dziesiętnie). W S-bloku S_1 w wierszu 2 i kolumnie 9 znajduje się wartość, 12 czyli wartością wyjściową z bloku S_1 jest wartość 1100 binarnie.

Poniżej przedstawiono wartości w poszczególnych S-blokach.

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

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

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

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

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

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

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

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

Po dokonaniu podstawień przy użyciu S-bloków wyjściowe 32 bity poddawane są permutacji P. Permutacja ta przebiega według następującej tabeli:

Permutacja 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

Nieodłącznym parametrem funkcji F jest klucz. Jest on inny dla każdego z 16 cyklów.

Poniżej przedstawiono schemat blokowy procedury tworzenia kluczy:

Opis tworzenia kluczy.

Długość klucz w algorytmie DES wynosi 56 bitów. Każdy cykl posiada natomiast własny klucz K_i składający się z 48 bitów. Pierwszą operacją przy tworzeniu kluczy jest dokonanie permutacji PC 1 według poniższej tabeli:

Permutacja PC 1						
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
63	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

Następnie dokonywany jest podział 56 bitów na dwie równe części C_0 i D_0 (każda po 28 bitów). Kolejnym krokiem jest przesunięcie bitów w lewo o określoną liczbę pozycji. Dla cyklów: i=1,2,9,16 jest to przesunięcie o jedną pozycję, dla pozostałych o dwie. Otrzymane w wyniku przesunięcia dane zapisywane są w C_i oraz D_i (gdzie i jest numerem cyklu) następnie dane te poddawane są permutacji PC2. Po dokonaniu tej permutacji otrzymujemy 48 bitowy klucz K_i (czyli klucz dla i-tego cyklu).

Permutacja PC 2					
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	32

Aby otrzymać klucz K_{i+1} należy korzystając z danych C_i oraz D_i dokonać przesunięcia w lewo o określoną liczbę pozycji. Otrzymane dane zapisać do C_{i+1} oraz D_{i+1}, a następnie dane te poddać permutacji PC2 itd.

Aby można było odszyfrować dane zakodowane algorytmem DES należy zastosować dokładnie ten sam algorytm, który stosuje się podczas szyfrowania z tą jednak różnicą, iż klucze należy użyć w odwrotnej kolejności. Zatem zamiast K 1 należy najpierw zastosować klucz K_{16}, potem K_{15} i tak dalej aż do K_1.

2. Plan wykonania ćwiczenia

1. Napisać program umożliwiający szyfrowanie plików z zastosowaniem algorytmu DES.

2. Sprawdzić poprawność działania programu poprzez zakodowanie pliku tekstowego, a następnie jego odkodowanie.

3. Literatura.

1. **Dorothy Elizabeth, Robling Denning - Kryptografia i ochrona danych.**

2. **Andrzej Kierzkowski - Ochrona programow i danych w praktyce.**
   Gliwice, Helion, 1992

3. **Opis algorytmu DES. http://db.tigra-system.pl/art.php?id=6**

4. **DES (Data Encrytpion Standard)**
   http://www.ws-webstyle.com/cms.php/en/netopedia/bezpieczestwo_hacking/des_data_encryption_standard