Uvod v kriptografijo

1. Poiščite skrito vsebino v spodnjem sporočilu:

9	4	1	6	0	3							5	2	8	1	6	6
5	6	0	8	5	0							2	6	1	4	0	1
2	9	2	3	5	2							2	6	2	2	1	4
1	3	7	1	9	4							1	2	3	5	9	1
0	7	4	1	9	9							7	0	8	9	2	0
4	3	0	8	3	1	3	1	1		3	4	2	1	2	2	9	0	4
					6	0	2	7		2	0	3	2					
					1	0	4	2		8	0	7	8					
					7	2	8	1	1	1	5	3	0					
								8	2	8								
					3	0	0	2	1	0	8	1	2					
					1	7	1	6		4	2	2	7					
					2	0	1	7		6	4	5	7					
3	8	6	5	1	1	2	3	4		4	4	9	4	9	5	6	1	5
6	1	3	1	4	8								2	6	1	6	2	3
9	5	3	9	7	3								0	9	3	5	1	9
6	8	3	5	9	8								8	4	8	8	8	2
7	3	1	7	3	6								6	7	8	8	4	2
3	2	2	9	9	4								8	3	8	4	3	7

Pomoč:

- Andrejev križ
- risanje kvadratov lahko olajša delo

2. Dešifrirajte spodnje sporočilo:

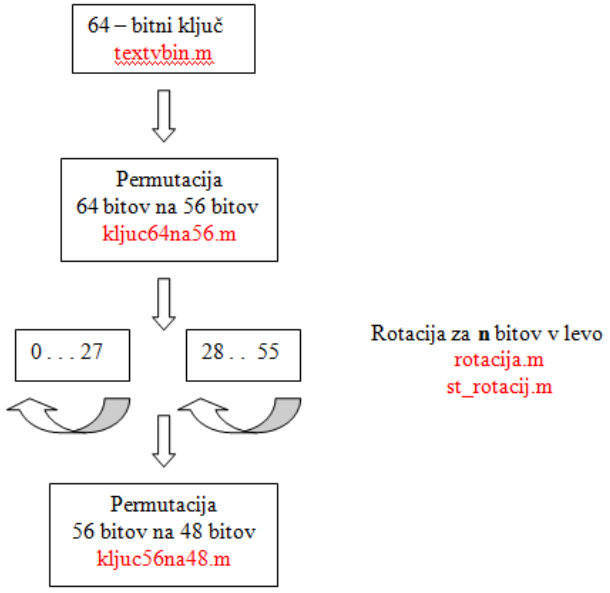
Detaoe obcfgkibo aogcno ftpetekbuk.

Pomoč:

- *klomonk*

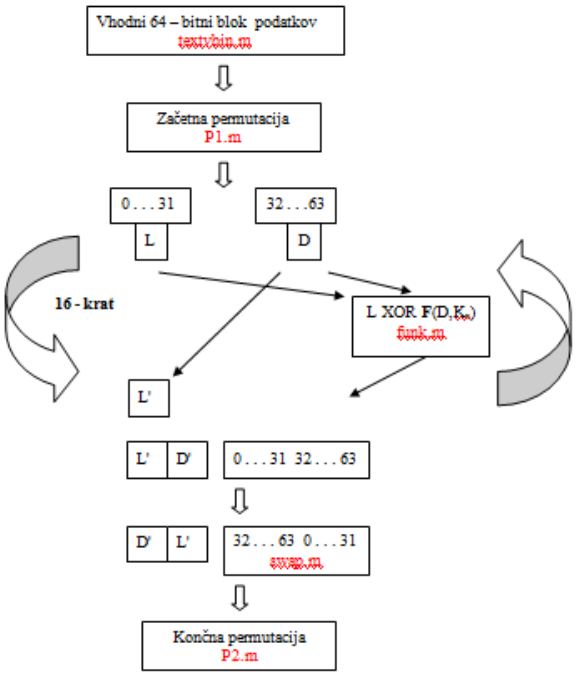
DES (*Data Encryption Standard*) šifriranje

3. V programskem okolju Matlab napišite funkcijo **DES(vhodniNiz, kljuci)** za šifriranje bloka 64 bitov v skladu z DES algoritmom:
    a. Z uporabo funkcije **textvbin()** pretvorite vhodni niz znakov v blok 64 bitov.
    b. Pripravite ključe:
        a. Za začetno permutacijo 64 -> 56 bitov uporabite funkcijo **kljuc64na56()**.
        b. Za 16 zaporednih rotacij in permutacij 56 -> 48 bitov uporabite funkciji **rotacija()** in **kljuc56na48().**

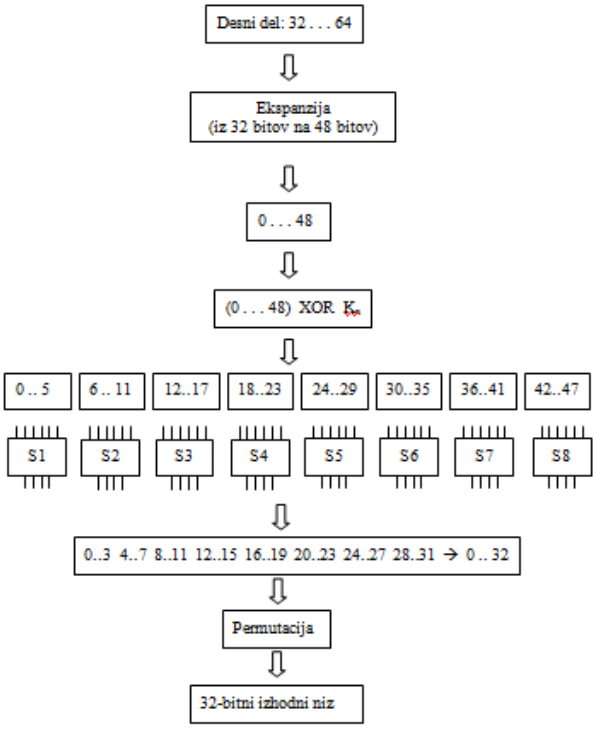


Slika 1: Priprava ključev.

    c. Prvo permutacijo bitov vhodnega niza opravite s funkcijo **P1()**.
    d. Razdelite blok bitov na levega in desnega.
    e. 16 zaporednih jedrnih xOR operacij opravite s funkcijo **xor()**. Po vsaki permutaciji je potrebno zamenjati levi in desni del bloka bitov.
    f. Pravilno združite oba dela bloka v izhodni končni blok bitov.
    g. Zadnjo permutacijo bitov vhodnega niza opravite s funkcijo **P2()**.
    h. Pretvorite končni blok bitov v niz znakov.



Slika 2: Jedro DES algoritma.



Slika 3: Funkcija F jedra DES algoritma.

```
function [izhod] = DES(vhodniNiz,kljuci)
% DES sifriranje
% vhodniNiz - niz znakov ne daljsi od 8
% kljuci - niz znakov ne daljsi od 8

% Pripravimo 64 bitne bloke vhodnih podatkov
vhod=textvbin(vhodni);

%% Pripravimo 64 bitne kljuce

kljuc=textvbin(kljuci);
kljuci=zeros(48,16);
tmp=kljuc64na56(kljuc);

for i=1:16
    tmp=rotacija(tmp,st_rotacij(i));
    kljuci(:,i)=kljuc56na48(tmp);
end

%% Sifriramo podatke

% Naredimo prvo P permutacijo nad vhodnimi podatki
tmp=P1(vhod);

% Vhodne podatke razbijemo na levi in desni blok
levi = tmp(1:32);
desni = tmp(33:64);

% 16-krat ponovimo jedrno XOR operacijo
for i=1:16
    tmplevi=desni;
    tmpdesni=xor(levi,funk(desni,kljuci(:,i)));
    levi=tmplevi;
    desni=tmpdesni;
end

% Zdruzimo bloka v en 64 bitni blok
rez=[levi;desni];

% Naredimo menjavo levega in desnega dela
rez1=swap(rez);

% Naredimo zadnjo P permutacijo nad vhodnimi podatki
konec=P2(rez1);
izhod=binvtext(konec);
```

Slika 4: Sintaksa funkcije DES(vhodniNiz, kljuci) za šifriranje v Matlabu.

4. Z namenom preizkusa delovanje funkcije *DES(vhodniNiz, kljuci)* šifrirajte besedo 'cistopis' s ključem '12345678':

   klic funkcije iz ukaznega okna okolja Matlab:

   DES('cistopis','12345678')

5. Preverite koliko bitov šifropisa se spremeni s spremembo enega bita čistopisa. V ta namen z uporabo funkcije **DES(vhodniNiz, kljuci)** šifrirajte besedi 'cistopis' in 'Cistopis' s ključem '12345678' in primerjajte rezultata.

   klic funkcije iz ukaznega okna okolja Matlab:

```
sifropis1=DES('cistopis','12345678');
sifropis2=DES('Cistopis','12345678');
biti1= textvbin(sifropis1);
biti2= textvbin(sifropis2);
steviloRazlicnihBitov=primerjajBite(biti1,biti2)
```

6. Funkciji **DES(vhodniNiz, kljuci)** dodajte funkcionalnost za dešifriranje šifropisa v skladu z DES algoritmom:
   i. Z uporabo **funkcije textvbin()** pretvorite niz znakov šifropisa v blok 64 bitov.
   j. Pripravite ključe
      a. Za začetno permutacijo 64 -> 56 bitov uporabite funkcijo **kljuc64na56()**.
      b. Za 16 zaporednih rotacij in permutacij 56 -> 48 bitov uporabite funkciji **rotacija()** in **kljuc56na48()**.

      Vrstni red ključev je pri dešifriranju natanko obraten kot pri šifriranju!

   k. Prvo permutacijo bitov šifropisa opravite s funkcijo **P1()**.
   l. Razdelite blok bitov na levega in desnega.
   m. 16 zaporednih jedrnih xOR operacij opravite s funkcijo **xor()**. Po vsaki permutaciji zamenjajte levi in desni del bloka bitov.
   n. Pravilno združite oba dela bloka v izhodni končni blok bitov.
   o. Zadnjo permutacijo bitov vhodnega niza opravite s funkcijo **P2()**.
   p. Pretvorite končni blok bitov v niz znakov.

```
function [izhod] = DES(vhodni,kljuci,enkripcija)
% DES sifriranje
% vhodniNiz - niz znakov ne daljsi od 8
% kljuci - niz znakov ne daljsi od 8
% primer klica funkcije - sifropis=DES('Janez','Micka',1);

% Pripravimo 64 bitne bloke vhodnih podatkov
vhod=textvbin(vhodni);

%% Pripravimo 64 bitne kljuce

kljuc=textvbin(kljuci);
kljuci=zeros(48,16);
tmp=kljuc64na56(kljuc);

for i=1:16
    tmp=rotacija(tmp,st_rotacij(i));
    if (enkripcija==1)
        kljuci(:,i)=kljuc56na48(tmp);  % ENKRIPCIJA
    end
    if (enkripcija==0)
        kljuci(:,17-i)=kljuc56na48(tmp);    % DEKRIPCIJA
    end
end

%% Sifriramo podatke

% Naredimo prvo P permutacijo nad vhodnimi podatki
tmp=P1(vhod);

% Vhodne podatke razbijemo na levi in desni blok
levi = tmp(1:32);
desni = tmp(33:64);

% 16-krat ponovimo jedrno XOR operacijo
for i=1:16
    tmplevi=desni;
    tmpdesni=xor(levi,funk(desni,kljuci(:,i)));
    levi=tmplevi;
    desni=tmpdesni;
end

% Zdruzimo bloka v en 64 bitni blok
rez=[levi;desni];

% Naredimo menjavo levega in desnega dela
rez1=swap(rez);

% Naredimo zadnjo P permutacijo nad vhodnimi podatki
konec=P2(rez1);
izhod=binvtext(konec);
```

Slika 5 Sintaksa funkcije DES(vhodniNiz, kljuci) za šifriranje in dešifriranje v Matlabu.

7. Sestavite ECB (*Electronic Code Book*) modul za DES šifriranje bitov n_{biti}>64.
8. Ugotovite vpliv dolžine ključa na varnost šifriranja. Dešifrirajte šifropis *sifropis* shranjen v datoteki ***skrivnost.mat***. Slednji je šifriran v skladu s postopkom DES, prvih pet bitov uporabljenega ključa je neznanih, ostalih 59 je enako 0.