
LABORATORIJSKA VEŽBA BR. 4

Simetrična enkripcija

CILJ VEŽBE

- Upoznavanje sa savremenim algoritmima kriptovanja
- Testiranje rada simetričnog algoritma DES
- Testiranje rada algoritma šifrovanja trostruki DES
- Testiranje rada simetričnog algoritma AES
- Testiranje rada simetričnog algoritma IDEA
- Testiranje rada simetričnog algoritma RC4
- Testiranje rada simetričnog algoritma RC6
- Testiranje rada simetričnog algoritma TWOFISH
- Upoznavanje sa programom za čuvanje podataka Truecrypt

POTREBNA OPREMA

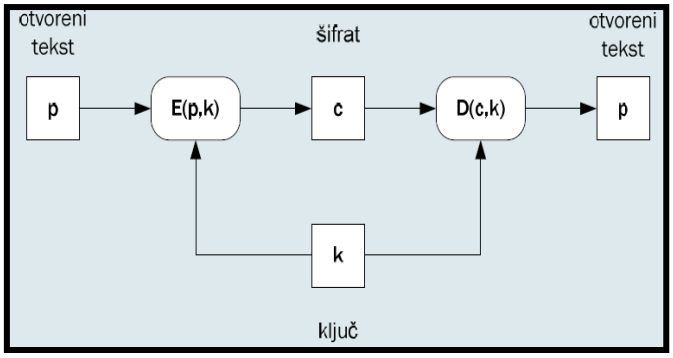
- Računar sa instaliranim Windows operativnim sistemom
- Istalirani programski paket Cryptool

TEORIJSKE OSNOVE

Savremeni kriptografski algoritmi

Pojava računara uslovila je razvoj novih algoritama kriptografije koji su svoje principe kriptovanja podataka zasnivali na snažnim računarskim karakteristikama računara. Metode klasične kriptografije zasnivale su se na tajnom pisanju, odnosno različitim matematičkim metodama koje su po nekom algoritmu primenjivane na otvorenom tekstu i na taj način ga činili nečitljivim trećim licima. Za razliku od njih metode moderne kriptografije svoj rad zasnivaju na tajnosti ključa preko koga se poruka može šifrovati i dešifrovati. U modernoj kriptografiji dakle važnija je **tajnost ključa** od tajnosti metode kriptovanja. U zavisnosti kako funkcionišu ključevi u kriptosistemu sve moderne sisteme kriptovanja tj. algoritme šifriranja možemo podeliti na:

1. **Simetrične algoritme** – koriste jedan odnosno privatni ključ
2. **Asimetrične algoritme** - koriste dva različita (javni i privatni) ključ.



Slika 1. *Blok šema principa rada simetričnog algoritma*

Simetrični algoritmi (vidi sliku 1.) za šifrovanje poruke **p** koriste ključ "**k**" kako bi dobili šifrat **c**. Dešifrovanje je obrnuto od prethodnog procesa, šifrat **c** se pomoću istog ključa "**k**" pretvara u originalnu poruku **p**. Osnovna osobina ovih algoritama je njihova brzina šifriranja pa su zato oni jako primenjivi za šifrovanje velikih datoteka. Ovi algoritmi se još nazivaju algoritmi sa jednim ključem (s*ingle key algorithms, one key algorithms*). Snaga ovih algoritama leži u tajnosti ključa. Dok god imamo potrebu da podatke šaljemo u šifrovanom obliku (što obično znači da treba da ostanu tajna za ostatak sveta), ključ za šifrovanje moramo držati u strogoj tajnosti (jer u suprotnom šifrovanje je totalno besmisleno). Šifrovanje i dešifrovanje se obavlja sledećim jednačinama:

$$E(K,P)=C$$
$$D(K,C)=P.$$

Simetrične algoritme možemo podeliti u dve grupe:

• **sekvencijalni algoritmi** (**protočni algoritmi** ili **algoritmi toka)** – šifruju poruku bajt po bajt.
• **blokovski algoritmi** ili **blokovske šifre** – šifruju delove teksta koji se nazivaju blokovi (npr, jedan blok je deo poruke dužine 64 ili 128 bita).

Režimi rada ECB i CBC

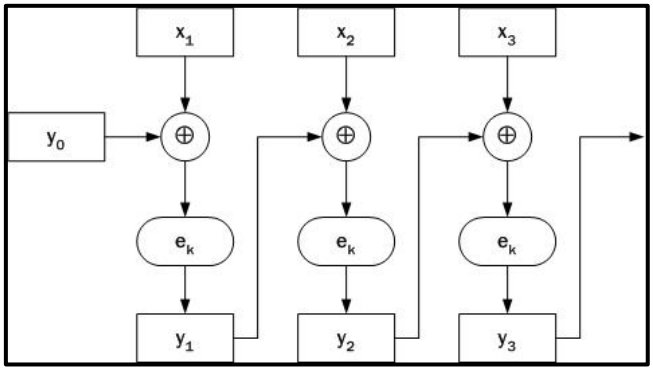
Pre nego što počnemo da testiramo rad simetričnih algoritama trebalo bi da napomenemo da postoje nekoliko režima rada ovih algoritama. Kao što je rečeno, blokovski algoritmi šifruju blok otvorenog teksta. To znači da DES šifruje blok dužine 64 bita. U realnim situacijama većina poruka je duža od 64 bita što znači da treba primeniti algoritam na više takvih blokova. Ovi režimi rada određuju način na koji se obavlja šifrovanje poruka dužih od jednog bloka.

Najjednostavniji režim rada je **ECB** (*electronic codebook mode*) – takozvani elektronski šifrarnik. Poruka se podeli na blokove dužine 64 bita (zadnji blok se dopuni do 64 bita slučajno generisanim nizom, ako je potrebno), a šifrovanje se obavlja blok po blok pomoću istog ključa. Identičnim blokovima otvorenog teksta odgovaraju identični blokovi šifrata. Jedan blok šifata zavisi samo od jednog bloka otvorenog teksta.

Prilikom šifrovanja u režimu **CBC** (režim ulančavanja blokova, *cipher block chaining*), najpre se računa rezultat XOR operacije izvršene nad trenutnim blokom otvorenog teksta i šifratom prethodnog bloka, a zatim se rezultat šifruje ključem K (videti sliku br.2). Povratna sprega postoji, blok šifrata zavisi od tekućeg i svih prethodnih blokova otvorenog teksta tako da identičnim blokovima otvorenog teksta u opštem slučaju odgovaraju različiti šifrati.

Vrednost *y0* je inicijalna vrednost (inicijalizujući vektor, IV) koja mora biti poznata i primaocu i pošiljaocu.Za šifrovanje i dešifrovanje koriste se sledeće relacije:

$$y_i = ek\ (y_{i-1}\ XOR\ x_i)\ za\ i \geq 1.$$
$$x_i = y_{i-1}\ XOR\ dk\ (y_i).$$

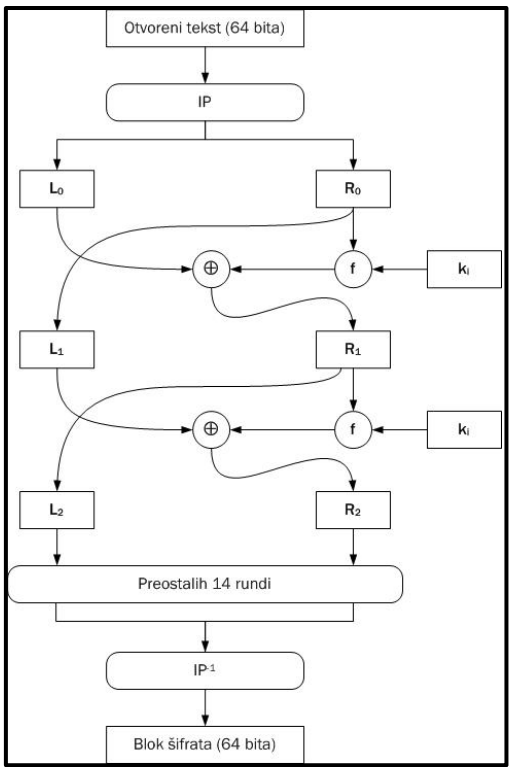


Slika 2*. CBC*

Feistelova mreža

Feistelova mreža (*Feistel network*) je simetričan blokovski algoritam koji u i-toj rundi obavlja operacije:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1}\ XOR\ f(R_{i-1}, k_i)$$



Slika 3. *Feistelova mreža*

Na slici br.3 prikazana je blok šema funkcionisanja *Feistelovog* algoritma. Parametar ***ki*** je takozvani podključ (vrednost koja se za svaku sledeću rundu nekim matematičkim

postupkom generiše na osnovu ključa). Funkcija ***f*** je funkcija runde tj neki matematički postupak koji pomoću podključa transformiše ulaz u izlaz. To znači da se blok podatka deli na dva dela, od kojih se jedan propusti kroz određenu funkciju, a drugi se dalje prenosi neizmenjen. Blokovi zatim zamene mesta, pa se obavi sledeća runda (broj rundi zavisi od algoritma).

1. **DES**

DES je simetričan algoritam koji šifruje tekst u blokovima dužine 64 bita, koristeći ključ dužine 56 bitova. Tako se dobija šifrat dužine 64 bita koji je duži od ključa za 8 bitova koji predstavljaju bitove parnosti svakog okteta(8 bitova). Tri osnovna koraka u algoritmu su:

1. **inicijalna permutacija IP**: $x_0 = L_0R_0 = IP(x)$ - gde je x blok otvorenog teksta, x_0 rezultat , L_0 i R_0 predstavljaju 32 viših i nižih bitova u x_0 respektivno. Obavlja se pomoću fiksne tablice za permutaciju.
2. **16 rundi** obrade podataka (proširenje, XOR sa ključem, supstitucija) - ulaz u prvu rundu je izlaz inicijalne permutacije. Ovaj deo algoritma je Feistelova mreža sa funkcijom f i podključevima koji se generišu na osnovu ključa.
3. **završna inverzna permutacija IP-1**. Transformiše izlaz iz poslednje runde u šifrat dužine 56 bita.

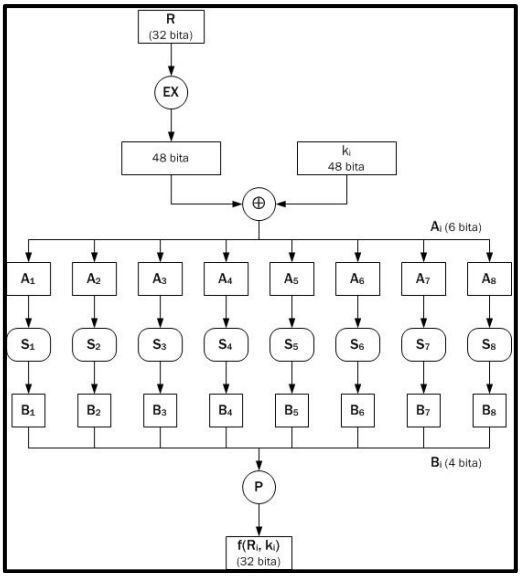
Za dešifrovanje DES šifrata koristi se isti algoritam kao i za šifrovanje. Polazi se od šifrata y, ali se potključevi koriste u obrnutom redosledu: k16, k15, ... , k1.

Funkcija f

Funkcija ***f*** prihvata dva ulazna argumenta: niža 32 bita izlaza iz prethodne runde (R_{i-1}) i potključ dužine 48 bitova (k_i). Kao rezultat se dobija niz dužine 32 bita. Funkcija se računa u sledeća 4 koraka:

1. Niz R_{i-1} proširuje se do niza dužine 48 bitova prema fiksnoj funkciji proširenja EX. Rezultujući niz $EX(R_{i-1})$ sastoji se od 32 bita niza R_{i-1}, permutovanih na određeni način, s tim što se 16 bitova pojavljuje dvaput.
2. Zatim se računa vrednost $A = EX(R_{i-1}) \text{ XOR } k_i$ a rezultat se zapisuje u obliku osam 6-bitnih nizova: $A = A_1\ A_2\ A_3\ A_4\ A_5\ A_6\ A_7\ A_8$.
3. Vrednost A se menja pomoću takozvanih S-boksova, tj. supstitucijskih kutija (*supstitution box*, S-box). Svaki S-box S_j (S_1, S_2, ... , S_8) predstavlja fiksnu matricu dimenzija 4×16, čiji su elementi celi brojevi između 0 i 15. Za dati niz od 6 bitova, $A_j = a_1\ a_2\ a_3\ a_4\ a_5\ a_6$, rezultat supstitucije $S_j(Aj)$ računa se na sledeći način:
    a. Dva bita $a_1\ a_6$ određuju binarni zapis reda ($0 \leq \text{red} \leq 3$) u S_j
    b. Četiri bita $a_2\ a_3\ a_4\ a_5$ određuju binarni zapis kolone ($0 \leq \text{kol} \leq 15$) u S_j.
    c. $B_j = Sj(A_j) = S_j(\text{red,kol})$, zapisano kao binarni broj dužine 4 bita.

    Na ovaj način se određuje $B = B_1B_2...B_8 = S_1(A_1)S_2(A_2)...S_8(A_8)$.
4. Niz bitova B dužine 32 bita permutuje se pomoću fiksne završne permutacije P. Tako se dobija P(B), tj $f(R_{i-1}, k_i)$.



Slika 4. *Funkcija f (DES)*

Generisanje podključeva

Ključ dužine 56 bitova, koji se koristi prilikom šifrovanja, čuva se u obliku K, dužine 64 bita. Bitovi parnosti na pozicijama 8, 16, 24, 32, 40, 48, 56 i 64 definisani su tako da svaki bajt sadrži neparan broj jedinica. Ovi bitovi se ignorišu pri računanju tabele ključeva. Ostalih 56 bitova permutuje se pomoću fiksne permutacije PK_1. Zapisuje se $PK_1(K) = C0D0$, gde su C0 i D0 viših i nižih 28 bitova u PK1(K). Za i = 1, 2, ... , 16 računa se:

$$\mathbf{C_i = LS_i(C_{i-1})}$$
$$\mathbf{D_i = LS_i(D_{i-1})}$$
$$\mathbf{k_i = PK_2(C_iD_i)}$$

LS_i je ciklički pomeraj ulevo za jednu poziciju, ako je i=1, 2, 9 ili 16, a u svim ostalim slučajevima za dve pozicije. PK_2 je fiksna permutacija.

Ključevi koji se ne koriste

Neki DES ključevi značajno su nesigurniji od ostalih, pa se ne koriste. Ukupno ima 64 ključa koje ne treba koristiti. U te ključeve spadaju:

a. **Slabi ključevi**. Generišu jednake podključeve u svakoj rundi. Šifrovanje i dešifrovanje sa slabim ključem je identična operacija. Slabi DES ključevi (zapisani sa bitovima parnosti) su: 0101010101010101, 1F1F1F1F0E0E0E0E, E0E0E0E0F1F1F1F1 i FEFEFEFEFEFEFEFE.
b. **Delimično slabi ključevi**. Generišu samo dva različita potključa, od kojih se svaki koristi u po 8 rundi. Par ključeva je par delimično slabih DES ključeva ako je šifrovanje sa jednim ključem isto je što i dešifrovanje sa drugim. Postoji šest pari delimično slabih DES ključeva. Na primer: 01FE01FE01FE01FE i FE01FE01FE01FE01.
c. **Potencijalno slabi ključevi** koji generišu samo četiri potključa (48 ključeva).

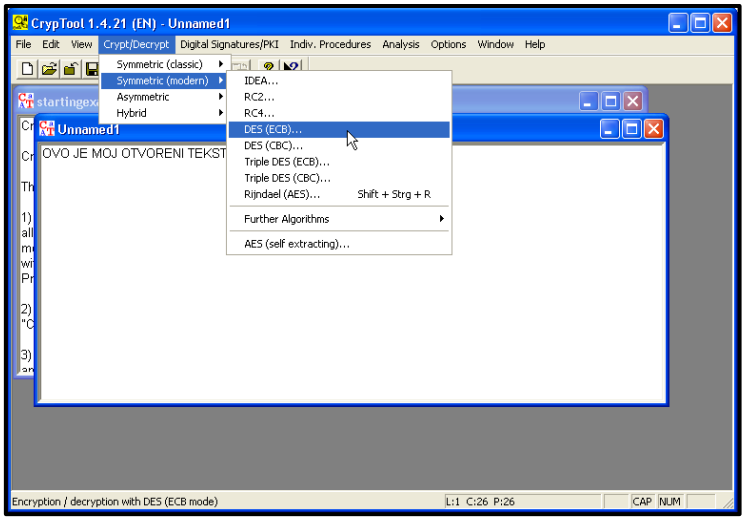
ZADATAK

Pomoću Cryptool-a šifrovati tekst "***OVO JE MOJ OTVORENI TEKST***" koristeći DES algoritam u ECB režimu pomoću ključa: 6A 9A AA F6 F9 FF EF FE. Dešifrujte dobijeni šifrat.

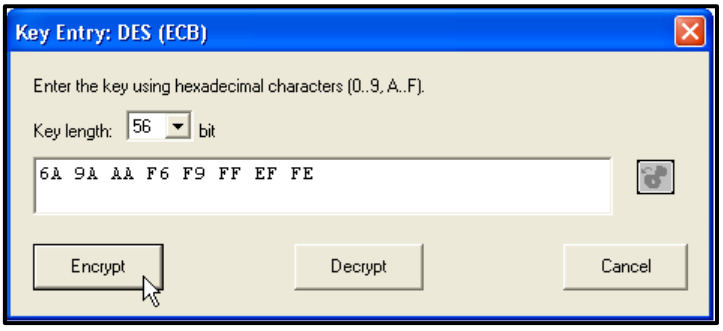
Zatim šifrujte isti otvoreni tekst DES algoritmom u CBC režimu istim ključem i uporedite ova dva dobijena rezultata šifrovanja. Da li su isti?

Postupak:

Otvoriti novi prozor za unos poruka: FILE → NEW.
Otkucati "OVO JE MOJ OTVORENI TEKST" u prozor za upis teksta.
Iz menija Crypt / Decrypt izabrati Symmetric (Moden) → DES (ECB).

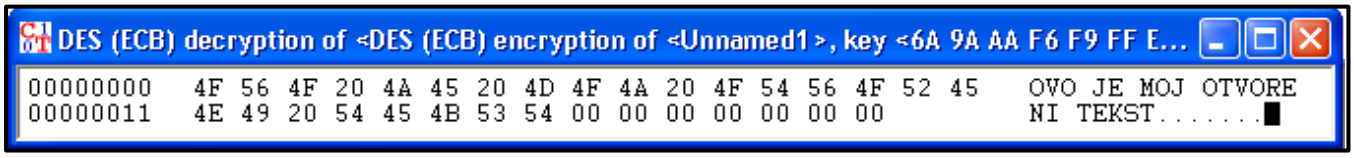


U polje za unos ključa kucate ključ iz teksta primera: 6A 9A AA F6 F9 FF EF FE. Klik na Encrypt.

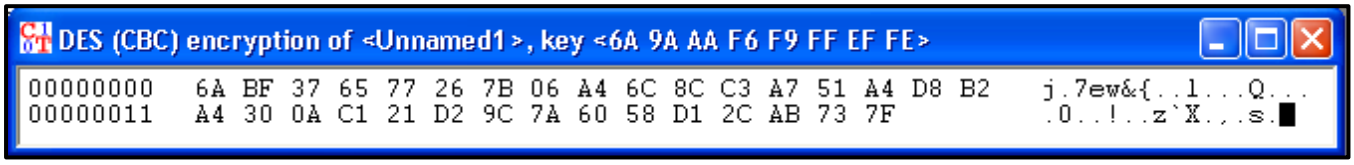


Dobićete sledeći šifrat.

Sada istim ključem odradite dešifrovanje dobijenog šifrata (postupak se razlikuje što ovde odradite klik na Decrypt). Trebalo bi da dobijete otvoreni tekst sa početka zadatka.



Sada ćete šifrovati isti otvoreni tekst, istim ključem, ali algoritmom DES koji radi u CBC režimu rada. Ne dobija se isti šifrat zato što je korišćen drugi režim rada (IV u startu menja šifrat a povratna sprega ostatak šifrata).



Dešifrujte ovo pomoću istog ključa i proverite da li se dobija početni otvoreni tekst.

2. TROSTRUKI DES

Šifrovanje i dešifrovanje dvostrukim DES algoritmom pomoću ključeva K_1 i K_2 definiše se na sledeći način:

$$\mathbf{y = eK_2\,(eK_1(x))}$$
$$\mathbf{x = dK_1\,(dK_2(y))}$$

Za razbijanje DES-a metodom grube sile potrebno je u najgorem slučaju ispitati 2^{112} ključeva. Međutim, broj operacija potreban za kriptoanalizu dvostrukog DES-a smanjuje se pomoću napada "**susret u sredini**" (*meet-in-themiddle*):

- Pretpostavimo da je poznat jedan par blokova otvoreni tekst – šifrat (x,y).
- Blok otvorenog teksta x redom se šifruje pomoću 2^{56} mogućih ključeva K_1, a rezultati se upisuju u tabelu i sortiraju po vrednostima $eK_1(x)$.
- Blok šifrata y redom se dešifruje pomoću 2^{56} mogućih ključeva K_2.
- Posle svakog dešifrovanja, u tabeli se traži rezultat $dK_2(y)$ takav da je $ek_1(x)=dk_2(y)$. Ukoliko se par ključeva (k_1,k_2) pronađe, testira se na poznatom paru otvoreni tekst – šifrat. Ukoliko ključevi zadovolje taj test, prihvataju se za korektne ključeve.

Potrebno je $2 \times 2^{56} = 2^{57}$ operacija (umesto 2^{112}) ali je tabela sa sortiranim ključevima jako velika. U svakom slučaju zbog smanjenja broja operacija potrebnih za napad dvostruki DES se NE koristi.
Trostruki DES se definiše na sledeći način:

$$\mathbf{x = dK_1\,(eK_2\,(dK_3\,(y)))}$$
$$\mathbf{y = eK_3\,(dK_2\,(eK_1\,(x)))}$$

Za trostruki DES, broj potrebnih operacija prilikom napada „susret u sredini" je reda veličine 2^{112}, što znači da je sigurnost trostrukog šifrovanja onakva kakvu smo očekivali od dvostrukog. Trostruki DES šifruje podatke pomoću ključa dužine 168 bitova (3x56) kako bi postigao sigurnost za koju je dovoljan ključ dužine 112 bitova. Pri tom obavlja operacije kroz 48 rundi (3x16) kako bi postigao sigurnost za koju je dovoljno 32 runde.

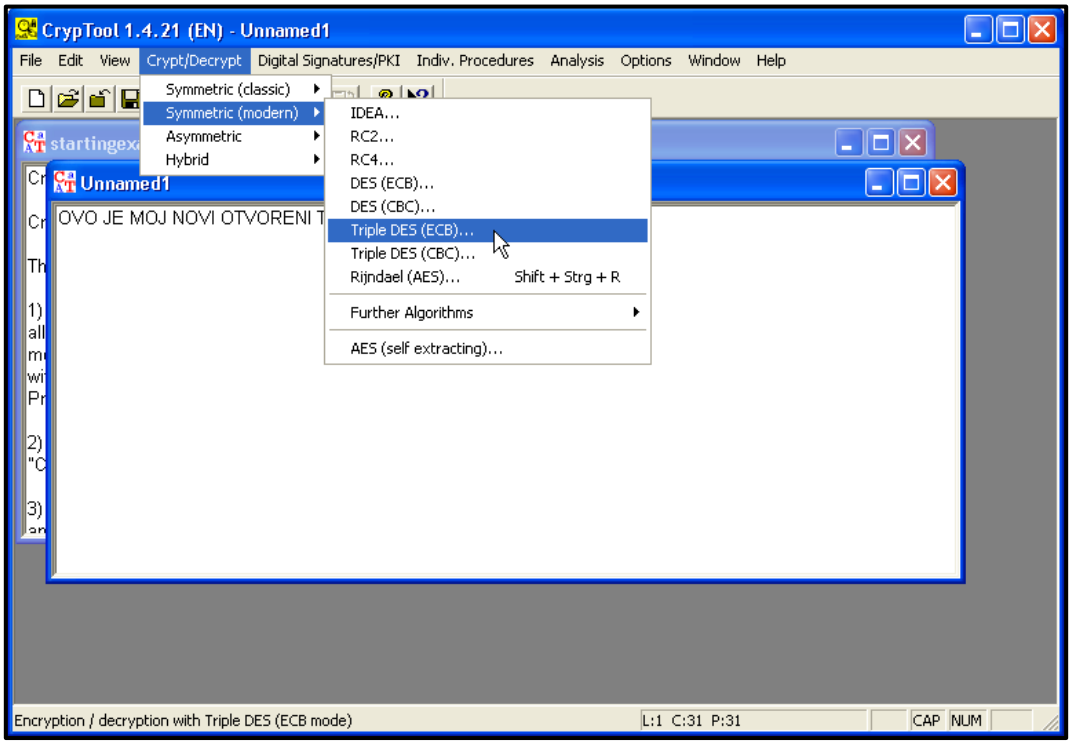
ZADATAK

Pomoću Cryptool-a šifrovati "OVO JE MOJ NOVI OTVORENI TEKST" koristeći algoritam Triple DES, u režimu rada ECB i ključ: 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 00. Šifrujte istu poruku u CBC režimu i uporedite šifrate.

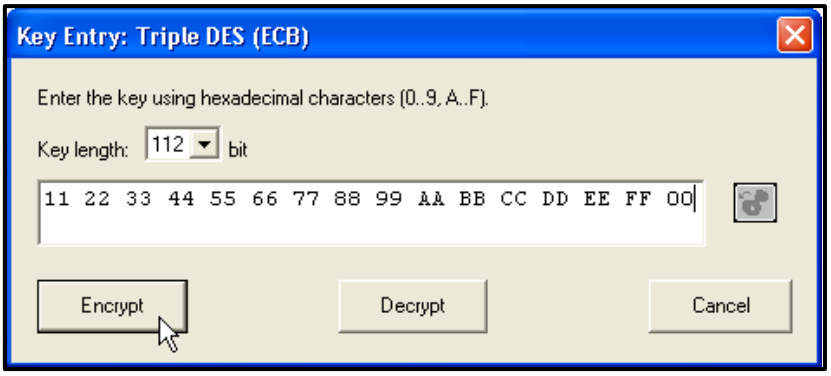
Postupak:

Otvoriti novi prozor za unos poruka: FILE → NEW.
Otkucati "OVO JE MOJ NOVI OTVORENI TEKST" u prozor za upis teksta.

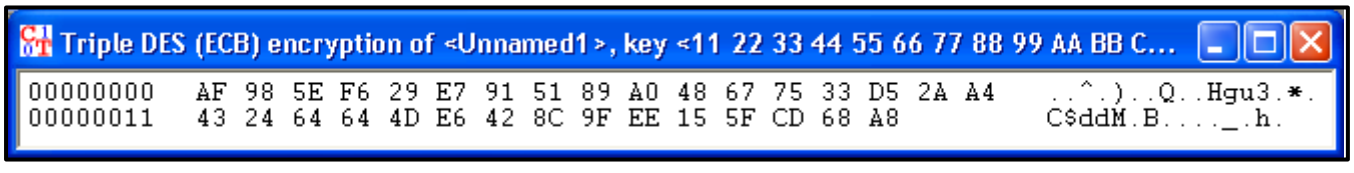
Iz menija Crypt / Decrypt izabrati Symmetric (Moden) → Triple DES (ECB).



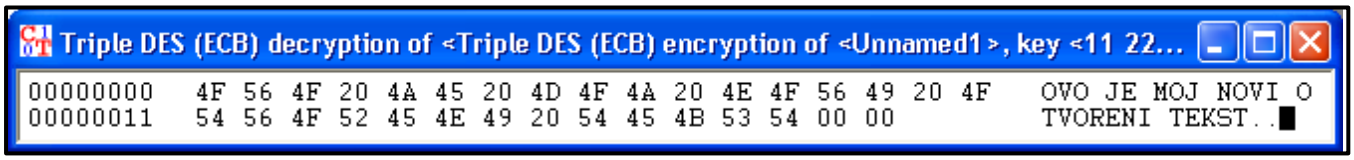
U polje za unos ključa kucate ključ iz teksta primera Klik na Encrypt.



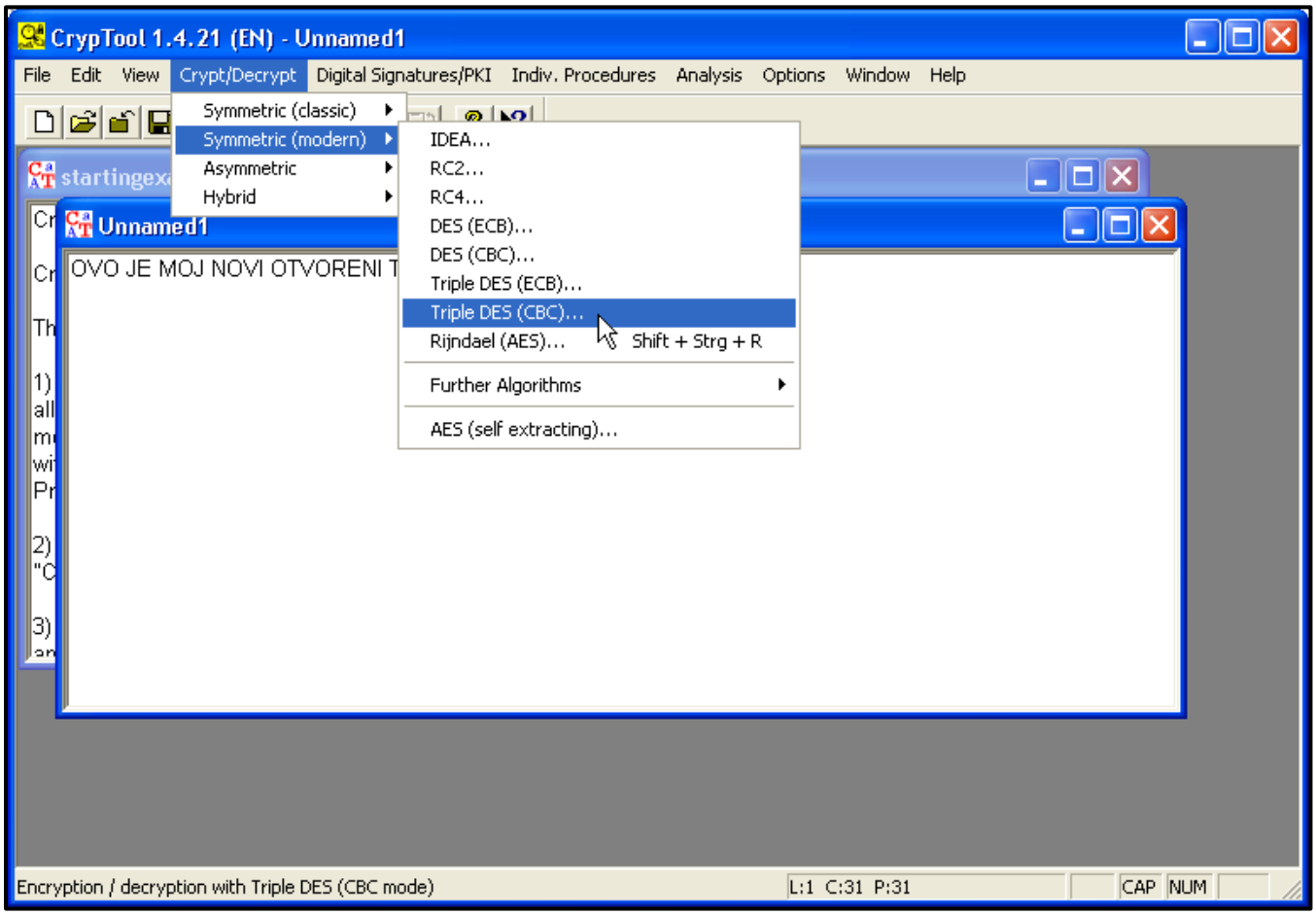
Dobićete sledeći šifrat.

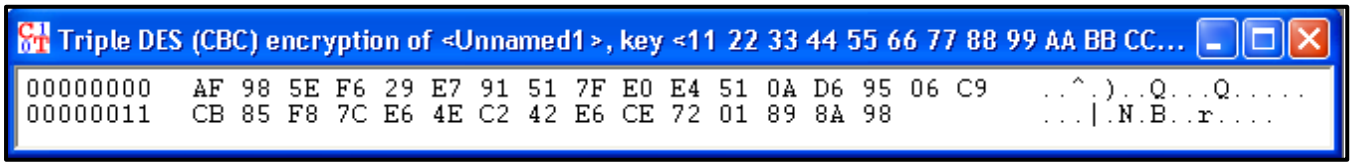


Sada istim ključem odradite dešifrovanje dobijenog šifrata (postupak se razlikuje što ovde odradite klik na Decrypt). Trebalo bi da dobijete otvoreni tekst sa početka zadatka.



Sada ćete šifrovati isti otvoreni tekst, istim ključem, ali algoritmom Triple DES koji radi u CBC režimu rada.





Uporedite šifrate u ECB i CBC režimima.
Dešifrujte ovo pomoću istog ključa i proverite da li se dobija početni otvoreni tekst.

3. AES

Algoritam koji su razvili Joan Daemen i Vincent Rijmen (RIJNDAEL algoritam po očetnim slovima njihovih prezimena) postao je novi standard za šifrovanje koji je dobio naziv – AES (*Advanced Encryption Standard*). Karakterističan je po tome da ne koristi Feistelove mreže (runde za izračunavanje šifrata se razlikuju od Feistelovih) kao i da prilikom konstrukcije supstitucijskih kutija koristi operacije u konačnom polju GF(2^8). Osnovne karakteristike RIJNDAEL algoritma su sledeće:

- veličina bloka za šifrovanje je promenljiva (128, 192 ili 256 bitova),
- dužina ključa je promenljiva (128, 192 ili 256 bitova),
- broj rundi je promenljiv i zavisi od dužne ključa i veličine bloka.

Matrica stanja je matrica bajtova koja opisuje trenutno stanje bloka za šifrovanje; dimenzije matrice zavise od veličine bloka za šifrovanje. Svaka ćelija predstavlja jedan bajt, tj. oktet. Broj redova matrice je fiksan (četiri reda), što znači da svaka kolona ima ukupno 32 bita,

tj. 4 okteta. Broj kolona matrice (N_w) zavisi od veličine bloka za šifrovanje i dobija se tako što se veličina bloka podeli sa 32. Na primer, blok podataka dužine 128 bita može se prikazati matricom stanja 4x4, 192-bitni blok matricom 4x6 a 256-bitni matricom 4x8.

Ključ za šifrovanje prikazuje se pomoću tabele sa četiri reda i N_k kolona, slično stanju bloka. Za 192-bitni ključ, tabela ključa će imati sledeći oblik 4x6.

Broj rundi ($N_{r)}$ je promenljiv, i zavisi od veličine bloka i dužine ključa. Konkretno, broj rundi je određen onim što je duže. Ako su i ključ i blok dužine 128 bitova ($N_w = N_k = 4$), onda je broj rundi 10. Ukoliko su ili ključ ili blok podatka dužine 192 bita ($N_w = 6$ ili $N_k = 6$), onda je broj rundi 12. Ako su ili ključ ili blok podatka dužine 256 bitova ($N_w = 8$ ili $N_k = 8$), onda je broj rundi 14. Što se AES standarda tiče, blok je uvek dužine 128 bitova, a ključevi mogu biti dužine 128 bitova (AES-128), 192 bita (AES-192) i 256 bitova (AES-256).

Sve operacije algoritma RIJNDAEL izvode se na matrici stanja. Nakon kopiranja ulaznog podatka, u matricu stanja se inicijalno dodaje potključ (*AddRoundKey*). Zatim se matrica stanja transformiše 10, 12 ili 14 puta, zavisno od dužine ključa, s tim da se poslednja runda transformacija razlikuje od prethodnih (izostavlja se transformacija *MixColumn*). Svaka runda algoritma predstavlja funkciju koja se sastoji od četiri transformacije nad oktetima:

- **ByteSub**. Zamena okteta na osnovu tabele supstitucije. Funkcija *ByteSub* je jedina nelinearna transformacija.
- **ShiftRow**. Pomeranje (rotiranje ulevo) okteta u redovima matrice stanja.
- **MixColumn**. Transformacije u svakoj koloni matrice stanja. Kolone se posmatraju kao polinomi četvrtog stepena sa koeficijentima iz GF(2^8) i množe sa konstantnim polinomom četvrtog stepena. *ShiftRow* i *MixColumn* formiraju linearni sloj koji obezbeđuju veliku difuziju bitova nakon nekoliko rundi.
- **AddRoundKey**. Dodavanje potključa u matricu stanja pomoću operacije ekskluzivno ILI.

Ključ se proširuje kako bi se generisali odgovarajući potključevi za operaciju *AddRoundKey*. Prve četiri reči proširenog ključa predstavlja ključ ***k***. Svaka sledeća reč se dobija izvođenjem operacije ekskluzivno ILI nad prethodnom reči i reči koja se nalazi $\mathbf{N_k}$ pozicija pre tekuće. Proširenjem ključa generiše se $\mathbf{(N_{r+1})N_w}$ 32-bitnih reči. Potključevi se biraju iz proširenog ključa tako da se prvi međuključ sastoji od prve četiri reči, drugi od sledeće četiri.

Dešifrovanje se obavlja pomoću istog algoritma, s tim što se u rundama koriste funkcije inverzne funkcijama *ByteSub*, *ShiftRow* i *MixColumn*, i nepromenjena funkcija *AddRoundKey*. Ukoliko algoritam za šifrovanje predstavimo sledećim pseudokodom:

```
stanje=ulaz;
AddRoundKey(stanje,w[0,Nw-1]);
for runda=1 to Nr-1 do {
	ByteSub(stanje);
	ShiftRow(stanje);
	MixColumn(stanje);
	AddRoundKey(stanje,w[runda*Nw,(runda+1)*Nw-1]);
}
ByteSub(stanje);
ShiftRow(stanje);
AddRoundKey(stanje,w[Nr*Nw,(Nr+1)*Nw-1];
izlaz=stanje;
```

Algoritam za dešifrovanje imaće sledeći oblik (funkcija *ByteSubInv* je inverzna funkciji *ByteSub*):

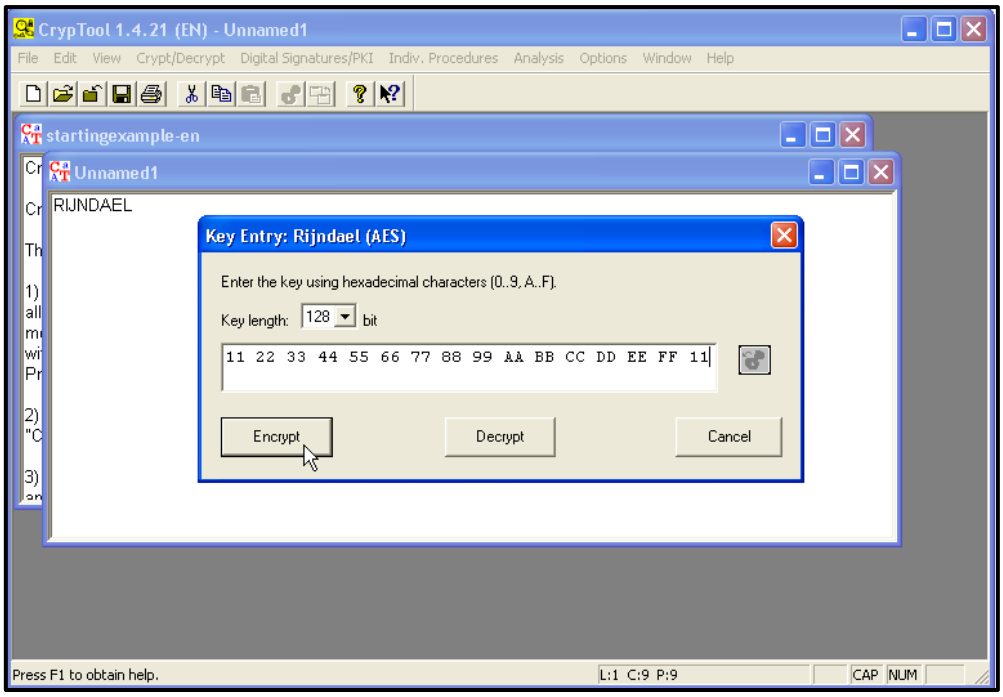
```
stanje=ulaz;
AddRoundKey(stanje,w[Nr*Nw,Nr+1)*Nw-1]);
for runda=Nr-1 to 1 do {
	ShiftRowInv(stanje);
	ByteSubInv(stanje);
	AddRoundKey(stanje,w[runda*Nw,(runda+1)*Nw-1]);
	MixColumnInv(stanje);
}
ShiftRowInv(stanje);
ByteSubInv(stanje);
AddRoundKey(stanje,w[0,Nw-1]);
```

ZADATAK

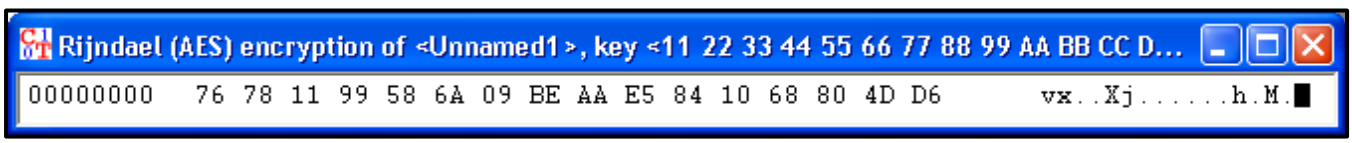
Pomoću Cryptool-a šifrovati reč RIJNDAEL koristeći algoritam AES/Rijndael i 128-bitni ključ: 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 11.

Postupak:

Otvoriti novi prozor za unos poruka: FILE → NEW.
Otkucati "RIJNDAEL" u prozor za upis teksta.
Iz menija Crypt / Decrypt izabrati Symmetric (Moden) → Rijndael (AES).
U polje za unos ključa kucate ključ iz teksta primera. Klik na Encrypt.



Dobijate šifrat:



Odradite sami dešifrovanje pomoću istog ključa.

Zadatak. Svaki student je dužan da pomoću gore objašnjenih algoritama šifruje **imena i prezimena** svojih članova porodice (Po četiri primera za svaki algoritam, ukoliko je broj članova porodice manji od četiri, studenti uzimaju **ime i prezime svog najboljeg prijatelja** kao četvrti primer).

Predmetni nastavnik i predmetni asistent