HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

DES algoritam

CCERT-PUBDOC-2003-06-24

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

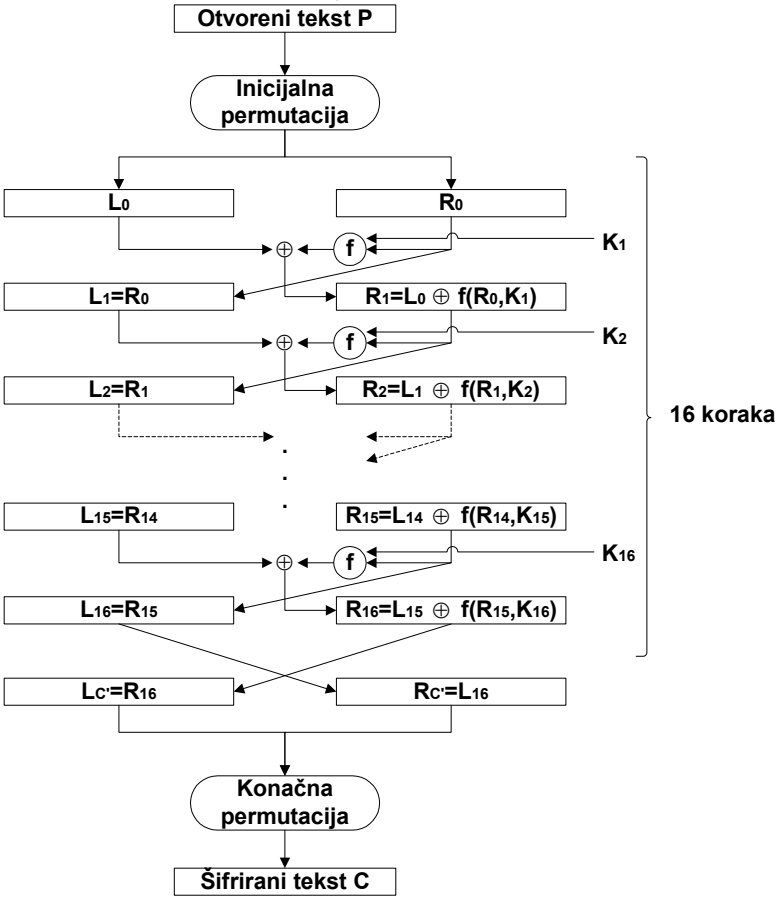
DES (*Data Encryption Standard*) algoritam je jedan od najpoznatijih i najraširenijih simetričnih algoritama. DES je razvijen tijekom 70-ih godina u IBM-u, te 1977. prihvaćen od Američke vlade kao standard za zaštitu podataka. Glavna mana mu je, što je javno poznato, da je tijekom razvoja mnoge "sugestije" davala i NSA (*National Security Agency*), pa mnogi smatraju da u algoritmu postoji sigurnosna "rupa" (engl. *backdoor*), koja omogućava američkoj vladi dešifriranje podataka. Bilo kako bilo, u svojom originalnom obliku, DES se tako i tako više ne smatra sigurnim, pošto se 56-bitni ključ u današnje vrijeme, i s raspoloživim računalnim resursima, može probiti i primjenom čiste sile (engl. *brute force*).

Iako potencijalno nesiguran, DES se i danas koristi za zaštitu podataka na mnogim instancama, a definiran je i kroz ANSI i NIST standarde. Također, mnogi prihvaćeni sigurnosni protokoli temelje svoj rad na DES algoritmu (SSL-TLS, IPSec).

2. Način rada

Otvoreni tekst (poruka) se šifrira u 64-bitnim blokovima, pri tom na izlazu dajući 64-bitni šifrirani tekst. Ključ za šifriranje je duljine 56-bita, no često se pojavljuje u 64-bitnom prikazu gdje se svaki osmi bit zanemaruje, odnosno može poslužiti npr. za provjeru pariteta.

Prvi korak algoritma jest inicijalna permutacija, dok posljednji korak predstavlja inverziju inicijalne permutacije. Prije posljednjeg koraka, lijeva i desna 32-bitna polovina se samo zamjenjuju. Kod preostalih 16 koraka, niz znakova se dijeli na lijevu i desnu 32-bitnu polovicu, gdje desna polovica postaje lijeva polovica idućeg koraka, dok se nad lijevom polovicom provode operacije koje su u svakom koraku parametrizirane drugim 48-bitnim podključem (ukupno ima 16 različitih 48-bitnih podključeva koji se izvode iz osnovnog ključa). Slika 1 daje shematski prikaz algoritma.



Slika 1: Shematski prikaz DES algoritma

Algoritam izgleda ovako:

```
početak
  permutiraj P
  P^T=L_0R_0
  za i=1 do i=16 radi
        L_i = R_i-1
        R_i = L_i-1 ⊕ f(R_i-1,S_i)
  kraj
  zamijeni(L_16,R_16)
  C' = L_16R_16
  C = permutiraj^-1C'
kraj
```

2.1. Inicijalna permutacija

Inicijalna permutacija može se opisati ovako:

```
početak
  za i=0 do i=7 radi
        P^T_i = P_58-i*8
        P^T_8+i = P_60-i*8
        P^T_16+i = P_62-i*8
        P^T_24+i = P_64-i*8
        P^T_32+i = P_57-i*8
        P^T_40+i = P_59-i*8
        P^T_48+i = P_61-i*8
        P^T_56+i = P_63-i*8
  kraj
kraj
```

Iako se može opisati na taj način, uobičajeno je da se permutacija prikaže u tabličnom obliku (Tablica 1). U tabličnom prikazu prvi bit se nalazi u gornjem lijevom uglu tablice, dok se posljednji bit nalazi u donjem desnom uglu (taj način prikaza odnosi se i na sve ostale tablice).

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

Tablica 1: Inicijalna permutacija

2.2. Konačna permutacija

Konačna permutacija može se opisati na sljedeći način:

```
početak
  za i=0 do i=3 radi
        C^T_16*i+1 = C'_40-i*2; C^T_16*i+2 = C'_8-i*2
        C^T_16*i+3 = C'_48-i*2; C^T_16*i+4 = C'_16-i*2
        C^T_16*i+5 = C'_56-i*2; C^T_16*i+6 = C'_24-i*2
        C^T_16*i+7 = C'_64-i*2; C^T_16*i+8 = C'_32-i*2
        C^T_16*i+9 = C'_39-i*2; C^T_16*i+10 = C'_7-i*2
        C^T_16*i+11 = C'_47-i*2; C^T_16*i+12 = C'_15-i*2
        C^T_16*i+13 = C'_55-i*2; C^T_16*i+14 = C'_23-i*2
        C^T_16*i+15 = C'_63-i*2; C^T_16*i+16 = C'_31-i*2
  kraj
kraj
```

Isto kao i kod inicijalne permutacije, uobičajeno je permutaciju prikazati u tabličnom obliku (Tablica 2).

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

Tablica 2: Konačna permutacija

2.3. Korak algoritma

Funkcija, koja je dio svakog koraka, prvo izvodi XOR operaciju nad odgovarajućim podključem i ekspandirana desna 32 bita prethodnog koraka. Tako dobivena 48-bitna vrijednost zatim se ponovno komprimira korištenjem 8 različitih S-blokova, koji za svaki 6-bitni ulaz daju 4-bitni izlaz. Tako dobiveni rezultat još jednom se permutira kroz P-blok da bi se zatim izvela još jedna XOR operacija s lijeva 32 bita prethodnog koraka.
Pošto se ključ pohranjuje u 64-bitnom obliku, prije generiranja podključeva potrebno ga je svesti na 56-bitni oblik. Taj postupak može se opisati na sljedeći način:

```
početak
  za i=0 do i=7 radi
        K_i = K'_57-i*8
        K_8+i = K'_58-i*8
        K_16+i = K'_59-i*8
        ako je i<4 onda
              K_24+i = K'_60-i*8
        inače
              K_48+i = K'_60-i*8
        K_28+i = K'_63-i*8
        K_36+i = K'_62-i*8
        K_44+i = K'_61-i*8
  kraj
kraj
```

I ova permutacija prikladnije se prikazuje u tabličnom obliku (Tablica 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
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

Tablica 3: Izvođenje 56-bitnog ključa iz 64-bitnog zapisa

Nakon što je 56-bitni ključ dobiven, on se dijeli na dva 28-bitna dijela, te se tako dobivene polovice kružno posmiču za 1 ili 2 bita, ovisno o koraku algoritma (Tablica 4).

Korak	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Posmak	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tablica 4: Posmak prilikom šifriranja

Zbog tih pomaka, različit podskup bitova originalnog ključa koristi se u svakom koraku. Iako se svaki od bitova koristi prosječno u 14 od 16 koraka, svi bitovi ključa ne koriste se jednako često.
Kako funkcija koja se računa pri svakom koraku algoritma koristi 48-bitne parametre, podključ koji se koristi mora biti komprimiran. Tablica 5 opisuje postupak kompresije.

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	31

Tablica 5: Kompresijska funkcija

Desna 32 bita, koja su isto tako parametar funkcije, shodno tome treba ekspandirati na 48-bitnu vrijednost. Način ekspanzije prikazan je u tablici (Tablica 6).

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

Tablica 6: Ekspanzijska funkcija

Ova ekspanzija, osim što služi da prilagođavanje duljini podključa prilikom izvođenja XOR operacije, ima i drugo, važnije, značenje. Naime, na način koji se ekspanzija provodi, omogućeno je da ulazni bitovi brže utječu na izlaz. DES je tako oblikovan da što brže zadovolji svojstvo da svaki bit šifriranog teksta ovisi o svakom bitu otvorenog teksta.

Nakon što je nad komprimiranim podključem i ekspandiranim blokom izvedena XOR operacija, izvodi se supstitucija korištenjem S-blokova. Svaki od 8 S-blokova (S_1 do S_8) ima 6-bitni ulaz, a kao rezultat daje 4-bitni izlaz (Slika 2). S-blokovi mogu se opisati tablicama s 4 retka i 16 stupaca unutar kojih se nalaze 4-bitni brojevi. Ukoliko se svaki ulaz u S-blokove označi bitovima b_1 do b_6, željeni redak tablice dobiva se kombiniranjem bitova b_1 i b_6 u 2-bitni broj, dok se željeni stupac tablice dobiva kombiniranjem bitova b_2 do b_5 u 4-bitni broj. Broj koji se nalazi na odgovarajućem mjestu u tablici predstavlja 4-bitni izlaz iz S-bloka (Tablica 7).

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

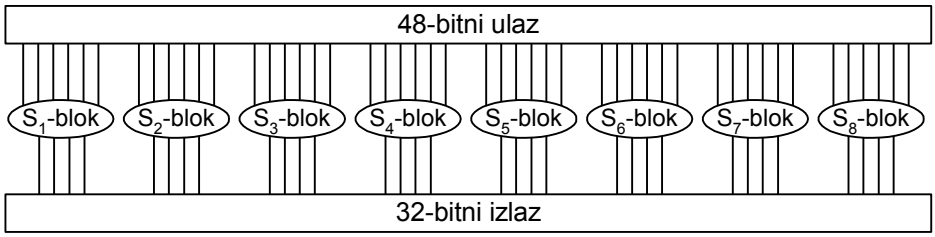
Tablica 7: Supstitucijske tablice za S-blokove

32-bitni rezultat supstitucije nakon toga se permutira kroz P-blok (Tablica 8).

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

Tablica 8: Funkcijska permutacija

Konačno, tako dobiveni rezultat se kombinira korištenjem XOR operacije s lijevim dijelom inicijalnog 64-bitnog bloka.



Slika 2: Supstitucija korištenjem S-blokova

2.4. Dešifriranje

Dešifriranje se provodi na identičan način, osim što se podključevi moraju primijeniti obrnuto, tako da ključ K_{16} postaje K_1, K_{15} postaje K_2 itd.
Isto tako, posmak prilikom rotacije ključa je malo izmijenjen (Tablica 9).

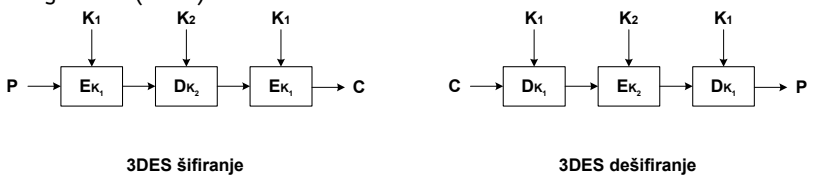
Korak	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Posmak	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tablica 9: Posmak prilikom dešifriranja

3. Inačice DES-a

3.1. 3DES

Kao što je već prije rečeno, u svojoj originalnoj formi DES se ne smatra pouzdanim, već samim time što je 56-bitna duljina ključa podložna napadu primjenom sile, no, pošto se algoritam i danas koristi za razne namjene, izvedene su razne modifikacije koje unapređuju sigurnost.
Jedno od takvih unapređenja jest i korištenje višestrukog DES algoritma. U praksi je najčešća uporaba trostrukog DES algoritma (3DES) u takozvanom EDE načinu rada.



Slika 3: Shematski prikaz 3DES algoritma

Slika 3 prikazuje trostruki DES algoritam u tzv. EDE načinu rada. Može se uočiti da 3DES algoritam koristi 112-bitni ključ za šifriranje (dva 56-bitna ključa). Pri postupku šifriranja koriste se tri modula; šifriranje, dešifriranje, te ponovno šifriranje (postupak dešifriranja kriptografski je identične složenosti kao i šifriranje). Implementacija 3DES algoritma, kako je prikazano na slici, podržava i korištenje standardnog DES algoritma korištenjem 56-bitnog ključa. U tom slučaju vrijedi da je $K_1=K_2=K$.

3.2. DESX

DESX je inačica DES algoritma koju su izradili u RSA Data Security. DESX se koristi tehnikom "izbjeljivanja" (engl. *whitening*), koja se temelji na uvođenju dodatnih XOR operacija nad ulaznim i izlaznim podacima u algoritmu. Kao dodatak 56-bitnom ključu, DESX koristi dodatni 64-bitni ključ koji se koristi za "izbjeljivanje". Ta 64 bita koriste se za izvođenje XOR operacije s otvorenim tekstom prije prvog koraka algoritma. Također, dodatna 64-bita izvedena korištenjem jednosmjerne funkcije nad ukupnim 120-bitnim DESX ključem koriste se za izvođenje još jedne XOR operacije u posljednjem koraku. Korištenjem ove modifikacije DES postaje otporniji na sve vrste napada.

3.3. Modifikacije S-blokova

Jedno od velikih pitanja kod DES algoritma su S-blokovi. Bilo je prijedloga da se alterniranjem S-blokova poboljša algoritam, no pokazalo se da je dizajn S-blokova, pa i njihov redoslijed, optimiran za napade korištenjem diferencijalne kriptoanalize. No, isto tako se pokazuje da njihov dizajn nije optimiran obzirom na napade korištenjem linearne kriptoanalize. Grupa istraživača imala je niz pokušaja promjena S-blokova kroz s^nDES algoritme, od kojih su neki bili uspješniji, dok su drugi pokazivali i lošije performanse prilikom linearne ili diferencijalne kriptoanalize (s^2DES, s^3DES).
Linearna i diferencijalna kriptoanaliza temelje se na poznavanju dizajna S-blokova. Ukoliko se S-blokovi naprave tako da budu ovisni o ključu i da se odabiru korištenjem neke kriptografski jake metode, primjena linearne ili diferencijalne kriptoanalize bila bi znatno otežana. Pri tome valja uzeti u obzir da slučajno odabrani S-blokovi, čak i ako su tajni, mogu imati vrlo loše diferencijalne i linearne karakteristike.

4. Sigurnost

Sigurnost DES algoritma upitna je s raznih aspekata. Upitna je duljina ključa, broj iteracija i dizajn S-blokova.
Najzanimljiviji pri tome su S-blokovi čija uporaba nikada nije u potpunosti razjašnjena. Iako IBM tvrdi da je uporaba baš takvih S-blokova posljedica godina i godina kriptoanalize, mnogi se plaše da je NSA ubacila *backdoor* kojim je omogućeno lagano dešifriranje poruka.
Također, način koji DES koristi da generiranje podključeva uzrokuje postojanje ključeva koji nisu prihvatljivi. Zbog toga što se glavni ključ dijeli na dva podključa, koji se zatim nezavisno posmiču, ključevi koji sadrže sve jedinice ili sve nule, te ključevi koji su pola jedinice a pola nule nisu sigurni, pošto induciraju postojanje identičnih podključeva u svakoj od iteracija algoritma. Isto tako, neki ključevi generiraju šifrirani tekst koji je identičan otvorenom tekstu. Nadalje, postoji skupina ključeva koji umjesto 16 različitih podključeva generiraju samo 2 različita podključa, a svaki od tih ključeva koristi se točno 8 puta u algoritmu (ukupno postoje 64 takva ključa). Također neki ključevi generiraju samo 4 podključa koji se onda koriste točno 4 puta u algoritmu (1116 ključeva). Obzirom da je ukupni broj mogućih kombinacija ključa 2^{56}, niti jedan od ovih slučajeva nije vjerojatan.
U originalnom prijedlogu algoritma predložena duljina ključa bila je 112 bita, no u konačnici taj ključ je skraćen na 56 bita. Takva duljina ključa otvara mogućnost napada primjenom sile, te kako je vrijeme protjecalo, a cijena računalnih resursa padala, pokazalo se da je uz adekvatna sredstava napad primjenom sile moguć i izvediv.
Postoje još i druge tehnike kriptoanalize koje olakšavaju napade na algoritam (npr. diferencijalna i linearna kriptoanaliza).

5. Zaključak

Već sama duljina DES ključa i struktura algoritma povlače mogućnost razbijanja u roku od nekoliko sati, pa i kraće, uz odgovarajuće računalne resurse. Uporabom 3DES algoritma ili drugih modificiranih inačica algoritma moguće je algoritam ipak učiniti dovoljno sigurnim od napada primjenom čiste sile, no uporaba algoritma u njegovom izvornom obliku nikako se ne može smatrati sigurnom.

Glasine o tajnom *backdooru* u algoritmu možda su istinite, možda i nisu, no primjena bilo koje modifikacije algoritma može otežati ili čak onemogućiti njegovo korištenje. U tom slučaju posebno je preporučljivo korištenje DES-a sa S-blokovima koji ovise o ključu.

Na kraju ostaje činjenica da je uporaba DES-a vrlo raširena (PKI, IPSec itd.), te da se u dogledno vrijeme, unatoč svim slabostima, ne očekuje njegova zamjena nekim drugim algoritmom. Isto tako, uz sve spomenute nedostatke, za osobne potrebe, pa čak i poslovne svrhe, DES algoritam s malim izmjenama se može smatrati dovoljno sigurnim.