Diferenčna kriptoanaliza

Uvod

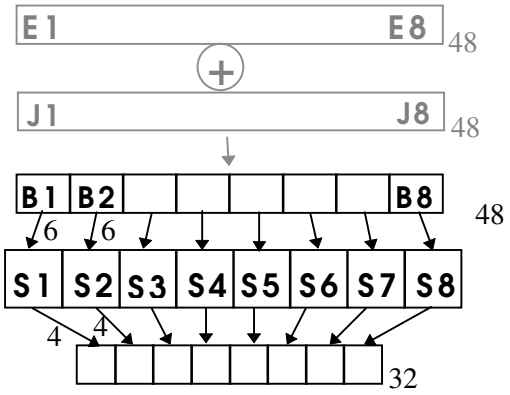
Diferenčno kriptoanalizo sta leta 1990 predstavila Eli Biham in Adi Shamir. Je nova metoda kriptoanalize in je bila dotlej javnosti neznana. Je metoda z izbranim čistopisom (chosen plaintext).
Diferenčna kriptoanaliza ima za osnovo izbrane pare čistopisa. Par čistopisa izberemo tako, da imata čistopisa izbrano diferenco (XOR). Oba čistopisa "spustimo" skozi cikle DES-a in opazujemo, kako se diferenca spreminja skozi posamezne cikle. Važna je diferenca dveh čistopisov, ne čistopisa sama. Z opazovanjem diference dobljenih tajnopisov priredimo različne verjetnosti za različne možne ključe. Z analiziranjem več parov čistopisa dobimo vedno večjo verjetnost za določen ključ.

Zaradi enostavnosti razlage DES poenostavimo; ignoriramo začetno permutacijo IP in končno inverzno permutacijo IP^{-1}.

Diferenčna kriptoanaliza bi bila učinkovita, če bi imel DES manj kot 16 ciklov. Če bi bilo ciklov 8, bi se dalo določiti ključ v nekaj minutah.

En cikel DES-a oz. lastnosti S-škatel

Za začetek opazujmo en cikel DES-a, oz. še manj. Opazujmo kaj se dogaja okoli S-škatel. Spustimo dva čitopisa v S-škatle; kakšne lastnosti veljajo za pripadajoča tajnopisa oz. njun XOR.



Slika1: S-škatle

Za določeno S-škatlo *Sj (1 <= j <= 8)* definirajmo urejeni par bitnih nizov dolžine 6: *(Bj,Bj*)*.
Diferenca (XOR) vhoda v *Sj* je: $Bj \oplus Bj^*$ (niz dolžine 6 bitov)
diferenca (XOR) izhoda iz *Sj* pa je: $Sj(Bj) \oplus Sj(Bj^*)$ (niz dolžine 4 bite)

Za vsako možno diferenco *Bj'* (vsak možni 6 bitni niz) definirajmo množico $\Delta Bj'$ vseh urejenih parov *(Bj,Bj*)* katerih XOR je diferenca *Bj'*: $Bj' = Bj \oplus Bj^*$ $(\Rightarrow Bj^* = Bj \oplus Bj')$

$\Delta Bj' = \{ (Bj,\ Bj \oplus Bj')\}$

Vseh možnih diferenc je $2^6 = 64$, torej imamo 64 množic $\Delta Bj'$. Vsaka množica $\Delta Bj'$ ima 64 elementov - urejenih parov *(Bj,Bj*).*

Primer: Lotimo se prve S-škatle (*S1*), diferenca vhoda v prvo škatlo (*B1'*) naj bo *110100*. Množica urejenih parov je potem taka:

Δ(110100) = { (000000, 110100), (000001, 110101)..... (111111, 001011)}

Za vsak urejeni par izračunamo diferenco izhoda iz *S1*. Npr. *S1(000000) = 1110* in *S1(110100) = 1001*, torej je difrenca izhoda *0111*. Če to naredimo za vseh 64 urejenih parov v množici *Δ(110100)*, pridemo do zelo zanimive porazdelitve rezultatov. Štiribitni nizi so zastopani zelo neenakomerno. Nekatere vrednosti so pogosto zastopane, nekatere pa sploh ne. Če opazujemo 64 parov čistopisa z diferenco *110100*, pripeljejo ti do naslednjih diferenc izhodov:

diferenca izhoda iz S1	0000	0001	0010	0011	0100	0101	0110	0111
št. urejenih parov s tako diferenco izhoda	0	8	16	6	2	0	0	12

1000	1001	1010	1011	1100	1101	1110	1111
6	0	0	0	0	8	0	6

Tebela 1: *N1(110100,C1')*

Pojavi se samo 8 od 16 možnih izhodnih vrednosti. Ta izbrani primer *Δ(110100)* izkazuje zelo neenakomerno porazdelitev. Če gledamo čez vse možnosti (za vsako škatlo *Sj* in vsak *Bj'*), se izkaže, da je povprečno zastopanih samo 75-80% možnih izhodov iz S-škatel.

Za vsak *1 <= j <= 8* (za vsako S škatlo) in za vsak možni *Bj'* (vsak možni niz dolžine 6 bitov) in *Cj'* (vsak možni niz dolžine 4 bite) definiramo:

INj (Bj',Cj') = { Bj : Sj(Bj) ⊕ Sj(Bj ⊕ Bj') = Cj'}

//Pri vhodni difrenci *Bj'* izračunamo množico parov *(Bj, Bj*),* ki imajo za rezultat *Cj'*, ker je *Bj** odvisen od *Bj'* ni potrebno opazovati parov, ampak samo *Bj.* Za *S1, B1' = 110100* naredimo te množice iz Tabele2.

Nj (Bj',Cj') = | INj (Bj',Cj') |

Število parov oz. *Bj*, ki ob izbranem *Bj'* pripeljejo do *Cj'.* Za *S1, B1' = 110100* razberemo te vrednosti iz Tabele1.

diferenca izhoda C1'	B1
0000	
0001	000011, 001111, 011110, 011111, 101010, 101011, 110111, 111011
0010	000100, 000101, 001110, 010001, 010010, 010100, 011010, 011011, 100000, 100101, 010110, 101110, 101111, 110000, 110001, 111010
0011	000001, 000010, 010101, 100001, 110101, 110110
0100	010011, 100111
0101	
0110	
0111	000000, 001000, 001101, 010111, 011000, 011101, 100011, 101001, 101100, 110100, 111001, 111100
1000	001001, 001100, 011001, 101101, 111000, 111101
1001	
1010	
1011	
1100	
1101	000110, 010000, 010110, 011100, 100010, 100100, 101000, 110010
1110	
1111	000111, 001010, 001011, 110011, 111110, 111111

Tabela 2: *Možni vhodi B1 za B1' = 110100, S1; IN1(110100,C1')*

Imamo 8 S-škatel, v vsako je možnih 64 diferenc vhodov (*Bj'*). To pomeni, da moramo narediti 512 tabel. Kar pa je z računalnikom enostavno narediti.

Do sedaj smo opazovali vhod neposredno v S-škatle *(B =B1B2...B8).* V ciklu DES-a pa je dejanski vhod v S-škatle *E* (E*(*R_{i-1}*)= E1E2...E8*; ekspanzija 32 bitov desnega dela sporočila), kateremu prištejemo po modulu 2 *J* (*J = Ki*; 48 bitov ključa za cikel i): *B = E ⊕ J*. Diferenca vhoda (za vseh osem S-škatel) se izrazi tako:

B ⊕ B = (E ⊕ J) ⊕ (E* ⊕ J) = E ⊕ E**

Pomemba je ugotovitev, da diferenca vhoda ni odvisna od ključa; vsekakor pa diferenca izhoda je odvisna od ključa.

Napišimo *B, E, J* kot 8 6-bitnih podnizov:

B =B1B2B3B4B5B6B7B8
E = E1E2E3E4E5E6E7E8E
J = J1J2J3J4J5J6J7J8J

in *B*, E*, J** na podoben način. Poznamo vse vrednosti *Ej* in *Ej** (dobljene so z ekspanizijo desnega dela čistopisa). Poznamo pa tudi vse diference izhoda *Cj' = Sj(Bj) ⊕ Sj(Bj*).* Potem mora veljati:

*Ej ⊕ Jj ∈ INj (Ej',Cj'); Ej' = Ej ⊕ Ej** // *INj(Bj',Cj') = INj(Ej',Cj');* ta lastnost
//omogoča določitev nabora možnih ključev

Definirajmo množico *testj:*

testj(Ej,Ej,Cj') = { Bj ⊕ Ej}* //nabor verjetnih 8 bitnih podključev
//Nabor verjetnih ključev dobimo tako, da vsakemu
//elementu množice *INj(Ej',Cj')* prištejemo *Ej*.

Kriptoanaliza poteka tako: dan imamo par *Ej, Ej*;* vidimo kakšen *Cj'* dobimo; pogledamo v usrezno tabelo *INj(Ej',Cj') = INj(Bj',Cj')* katere so možne vrednosti *Bj*. Možne ključe *testj* pa dobimo tako: *testj = Bj ⊕ Ej.* Samo en element množice *testj* je pravi ključ *Jj.*

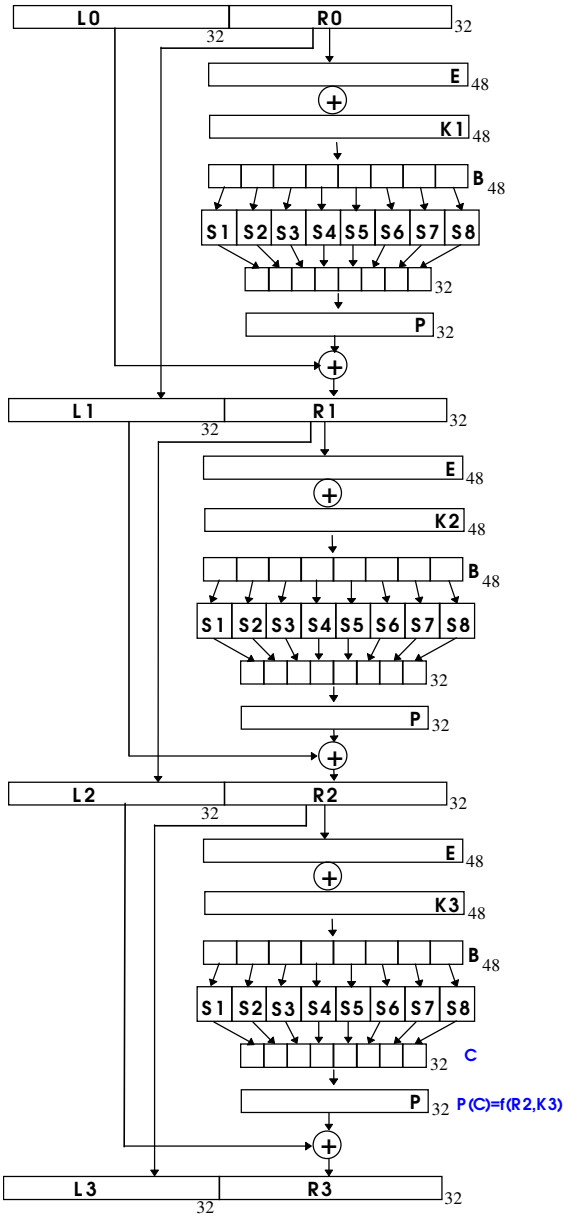
Primer:
Izberimo par čistopisa:
E1 = 000001 in
E1 = 110101 (⇒E1' = 110100).*
Za diferenco izhoda dobimo:
C1' = 1101.

N1(110100,1101) = 8, to pomeni, da 8 vrednosti *B1* pripelje do tega izhoda.
IN1(110100,1100) = { 000110, 010000, 010110, 011100, 100010, 100100, 101000, 110010 }
⇒ test1(000001, 110101, 1101) =
{ 000111, 010001, 010111, 011101, 100011, 100101, 01001, 110011}

Dobili smo 8 kandidatov za del ključa *J1*. Z drugim parom čistopisa bi dobili drugo množico možnih kandidatov za ključ *test1*. Kandidat za *J1* je v preseku obeh množič. S tretjim parom čistopisa bi še zmanjšali množico kandidatov...

Napad na DES s tremi cikli

Uporabimo ugotovitve iz prejšnjega poglavja za razbitje DES-a, če bi ta imel samo 3 cikle. Napadamo s premišljeno izbranim parom čistopisa.



Slika 2: 3-cikli DES-a

Para čistopisa: *L0R0, L0*R0**
para tajnopisa: *L3R3, L3*R3**

Izrazimo R3:

$$R3 = L2 \oplus f(R2, K3)$$
$$= R1 \oplus f(R2,K3)$$
$$= L0 \oplus f(R0,K1) \oplus f(R2,K3)$$

$$R3^* = L0^* \oplus f(R0^*,K1) \oplus f(R2^*,K3)$$

$$\Rightarrow \quad R3' = L0' \oplus f(R0,K1) \oplus f(R0^*,K1) \oplus f(R2,K3) \oplus f(R2^*,K3)$$

Izbrani čistopis naj bo tak, da *R0 = R0*,* torej:

$$R0' = 00...0$$

Zato velja $f(R0,K1) = f(R0^*,K1)$ in enačbo za *R3* poenostavimo:

$$R3' = L0' \oplus f(R2,K3) \oplus f(R2^*,K3)$$

R3' je znan, saj je desni del diference dobljenih tajnopisov, *L0'* je tudi znan, saj je levi del diference čistopisov. Torej lahko nastavimo enačbo za $f(R2,K3) \oplus f(R2^*,K3)$:

$$f(R2,K3) \oplus f(R2^*,K3) = R3' \oplus L0'$$

$f(R2,K3) = P(C)$ in $f(R2^*,K3)=P(C^*)$, kjer sta *C* in *C** izhoda iz osmih S-škatel. P je fiksna znana permutacija. Zato velja:

$$P(C) \oplus P(C^*) = R3' \oplus L0'$$

in naprej:

$C' = C \oplus C^* = P^{-1}(R3' \oplus L0')$ To je diferenca izhoda iz S-škatel v tretjem ciklu.

Znana sta tudi vhoda v tretji cikel *E* in *E*.* Izračunamo ju iz *R2 = L3* in *R2* = L3** (znani vrednosti; del dobljenega tajnopisa) z ekspanzijo E.

$E = E(L3)$
$E^* = E(L3^*)$

S skrbno izbranim čistopisom smo kriptoanalizo DES-a s tremi cikli reducirali na kriptoanalizo enega cikla DES-a. Znani imamo *E, E** in *Cj',* tako lahko kreiramo množice kandidatov za ključ:

test1 test2 test3 test4 test5 test6 test7 test8

test1 predstavlja nabor prvih možnih 8 bitov ključa *K3, test2* predstavlja nabor drugih možnih 8 bitov ključa *K3...*Nabor možnih ključev zmanjšamo tako, da izberemo nov par izbranega čistopisa in kreiramo nove množice kandidatov za ključe *test1...test8.* Izkaže se, da ni potrebno prav veliko parov čistopisov za določitev celotnega ključa. V primeru v knjigi smo prišli do ključa *J1...J8* že po treh izbraanih parih čistopisa.

Tako smo dobili 48 bitov ključa *K3*. *K3* dobimo iz *K* z znano transformacijo (key schedule), tako tudi 48 bitov ključa *K* dobimo iz *K3* na enostaven način. Ostalih 8 bitov ključa *K* dobimo tako, da pregledamo vse možnosti za neznanih 8 bitov ključa, ki jih je $2^8 = 256$ (exhaustive search).

Napad na DES s šestimi cikli

Napad na DES s šestimi cikli vpelje še določanje verjetnosti. Upoštevamo verjetnost, da se ob dani diferenci vhoda pojavi kot diferenca izhoda določena vrednost.
Premišljeno izberemo par čistopisa (s določeno diferenco) in določimo verjetnost pojavljanja željene vrednosti diference skozi cikle enkripcije.

Primer določanja verjetnosti za DES z enim ciklom.

izbrani XOR čistopisa:	$L0' = karkoli$	$R0' = 00000000_{16}$	
dobljeni XOR tajnopisa:	$L1' = 00000000_{16}$	$R1' = L0'$	$p = 1$

// $R0' = 00000000_{16} \Rightarrow R0 = R0^* \Rightarrow f(R0,K1) = f(R0^*,K1) \Rightarrow R1' = L0'$

Primer: DES z enim ciklom, Če imamo vhodno difreneco $L0'$=karkoli $R0'=00000000_{16}$, kakšna je verjetnost, da je izhodna diferenca $L1' = 60000000_{16}$ $R1' = 00808200_{16}$

izbrani XOR čistopisa:	$L0' = 00000000_{16}$	$R0' = 60000000_{16}$	
dobljeni XOR tajnopisa:	$L1' = 60000000_{16}$	$R1' = 00808200_{16}$	$p = 14/64$

// Ekspanzija $R0'$ ima za rezultat $E1' = 001100$, vhodi v ostale S-škatle so 000000. Diferenca izhoda iz škatel $S2...S8$ bo 0000. Diferenca izhoda iz škatle $S1$ bo 1110 z verjetnostjo $14/64$ (ker je $N1(001100,1110) = 14$). Tako dobimo:
$C' = 11100000000000000000000000000000$ z verjetnostjo $14/64$. Nato sledi še permutacija:
$P(C) \oplus P(C^*) = 00000001000000010000001000000000$ kar je v šestnajstiškem 00808200_{16}.
To je po XOR z $L0'$ tudi rezultat ($R1'$). Seveda je vedno $L1' = R0'$.

Napad na DES s šestimi cikli temelji na karakteristiki verjetnosti po 3 ciklih DES-a. Pri DES-u s šestimi cikli hočemo imeti vhodno diferenco v 4 cikel točno določeno vrednost. Štartamo z diferenco:

$L0' = 40080000_{16}$ $R0' = 04000000_{16}$

Izhodno difreneco po treh ciklih (= vhodna diferenca v četrti cikel) hočemo imeti:

$L3' = 04000000_{16}$ $R3' = 40080000_{16}$

Taka izhodna diferenca po treh ciklih se ob zgornji vhodni diferenci pojavi z verjetnostjo $1/16$:

$L0' = 40080000_{16}$	$R0' = 04000000_{16}$	
$L1' = 04000000_{16}$	$R1' = 00000000_{16}$	$p = 1/4$
$L2' = 00000000_{16}$	$R2' = 04000000_{16}$	$p = 1$
$L3' = 04000000_{16}$	$R3' = 40080000_{16}$	$p = 1/4$

Izrazimo $R6$:

$$R6 = L5 \oplus f(R5, K6)$$
$$= R4 \oplus f(R5,K6)$$
$$= L3 \oplus f(R3,K4) \oplus f(R5,K6)$$

$$R6^* = L3^* \oplus f(R3^*,K1) \oplus f(R5^*,K6)$$

$$\Rightarrow \quad R6' = L3' \oplus f(R3,K4) \oplus f(R3^*,K4) \oplus f(R5,K6) \oplus f(R5^*,K6)$$

$R6'$ je znan. $L3' = 04000000_{16}$ in $R3' = 40080000_{16,}$ ampak samo z verjetnostjo $1/16$. V tem posebnem primeru je vhodna diferenca v S-škatle v četrti cikel taka (po ekspanziji $R3'=40080000_{16}$):

$001000000000000001010000...0$

Vhodne diference v $S2$, $S5$, $S6$, $S7$ in $S8$ so 000000, zato so izhodne diference iz teh škatel v četrtem ciklu 0000. Enačba za $R6'$ se samo za te določene pozicije poenostavi:

$R6' = \quad L3' \oplus f(R5,K6) \oplus f(R5^*,K6)$

Upoštevajmo: $f(R5,K6) \oplus f(R5^*,K6) = P(C')$

$\Rightarrow \qquad C' = P^{-1}(R6' \oplus 04000000_{16})$
$C1'C2'C3'C4'C5'C6'C7'C8' = P^{-1}(R6' \oplus 04000000_{16})$

Z verjetnostjo *1/16* lahko rečemo, da so *C2', C5', C6', C7'* in *C8'* diference izhodov is *S2, S5, S6, S7* in *S8* v šestem ciklu. Vhodi v te S-škatle v šestem ciklu izračunamo iz dobljenega tajnopisa:

$E1E2E3E4E5E6E7E8 = E(R5) = E(L6)$
$E1^*E2^*E3^*E4^*E5^*E6^*E7^*E8^* = E(R5^*) = E(L6^*)$

Tako se problem določevanja 30 bitov ključa (poenostavljena enačba velja namreč samo za določene pozicije bitov) *J2, J5, J6, J7* in *J8* reducira na nalogo, enako določevanju ključa v napadu na DES s tremi cikli. Vendar obstaja še pomemben problem; hipotetična diferenca izhoda po šestem ciklu velja samo z verjetnostjo 1/16 (po tretjem ciklu nas je samo 1/16 vhodnih parov pripeljala do željenega vhoda v četrti cikel). V ostalih 15/16 primerov dobimo napačne naključne rezultate namesto možnih ključev. Nekako moramo določiti, kdaj dobimo prave možne kandidate za ključ. Najti moramo način, kako se znebiti ostalih (15/16) napačnih predlogov. Na prvi pogled izgleda to nemogoče, a obstaja način, kako se znebiti veliko možnih napačnih predlogov.

Prva strategija je,da za vse možne vhodne pare izračunanamo *Ej, Ej** in *Cj'* in kreiramo nabore ključev *testj*(*Ej, Ej* ,Cj'*) za *j = 2,5,6,7,8.* Če smo začeli s pravim parom, ne bomo imeli za noben *j* prazne množice *testj*. Za vsak podniz ključa bomo imeli ponujenih nekaj možnosti. Če pa začnemo z napačnim parom, bo *Cj'* napačen in *testj* bodo naključne vrednosti, z določeno verjetnostjo, da je nabor *testj* za kakšen *j* prazen. Če je $/ testj / = 0$ za katerikoli $j \in \{2,5,6,7,8\}$ imamo vsekakor napačen par. Verjetnost, da je $/ testj / = 0$ je 1/5 (glej izvajanje na strani 2). Verjetnost, da vseh 5 množic *testj* ni praznih, je torej $0.8^5 \approx 0.33$; torej je verjetnost, da je vsaj ena množica prazna okoli 0.67. Tako smo z operacijo filtriranja izločili približno 2/3 napačnih parov. Delež pravilnih parov je po operaciji filtriranja približno (1/16)/(1/3) =3/16.

Recimo, da par prestane operacijo filtriranja. Se pravi, da par ponudi za vsak $j \in \{2,5,6,7,8\}$ nabor *testj.* Iz teh naborov *testj* sestavimo toliko predlaganih ključev *J2J5J6J7J8* (seveda samo 30 bitov ključa):

$$\prod_{j \in \{2,5,6,7,8\}} = / testj /$$

To število predlaganih ključev je (že za en par!) lahko zelo veliko, npr. večje kot 8000.

Recimo, da se lotimo kriptoanalize z N pari (ki so že prestali operacijo filtriranja). Pišimo v tabelo vse predlagane ključe. Pravilni predlagani ključ se bo pojavil v 3*N/16 primerih! Nepravilni predlagani ključi se bodo pojavljali naključno (možno jih je 2^{30}), z mnogo manjšo verjetnostjo.

Tako smo določili 30 bitov ključa. Z izbiro vhodne diference $L0'=00200008_{16}$ $R0'=00000400_{16}$ lahko določimo ključa *J1* in *J4,* torej nadaljnih 12 bitov. Ostane še 14 bitov ključa, ki pa se jih že da določiti s tem, da poskusimo vse možnosti (exhaustive search).

Za napad na DES s šestimi cikli je dovolj že 120 parov izbranega čistopisa (še pred operacijo filtriranja).

Zaključek

Tabela prikazuje, koliko izbranih čistopisov potrebujemo za razbitje DES-a z različnim številom ciklov.

8	2^{14}
10	2^{24}
12	2^{31}
14	2^{39}
16	2^{47}

Na diferenčno kriptoanalizo so občutljivi tudi drugi algoritmi, ki vključujejo substitucije in permutacije npr. FEAL, REDOC-II in LOKI.