Capitolul 6

Atacuri asupra sistemelor de criptare bloc

6.1 Consideraţii generale

În acest capitol vom trece în revistă principalele tipuri de atac dezvoltate asupra sistemelor de criptare bloc. Majoritatea lor au fost create iniţial pentru criptanaliza sistemului DES, devenind ulterior standarde de securitate pentru toate sistemele de criptare. Tipuri de criptanaliză mai vechi – cum sunt criptanaliza diferenţială sau liniară – au generat ulterior modalităţi noi de atac (mixt, boomerang, rectangle etc) pentru anumite tipuri de criptosisteme, adesea semnjificativ mai puternice.

6.2 Compromisul spaţiu - timp

Este un tip de atac bazat pe un compromis între complexitatea spaţiu şi complexitatea timp a atacurilor prin forţă brută, care poate fi utilizat destul de eficient în unele atacuri cu text clar ales.

Aici $Oscar$ va dispune de o pereche (α, β) cu $\beta = e_K(\alpha)$, şi caută să determine cheia K.

Nu vom folosi o structură particulară de implementare a unui sistem, dar ne vom referi în mod implicit la sistemul DES; mai exact, ştim că textele clare şi cele criptate sunt de 64 biţi, iar cheia are 56 biţi.

O căutare exhaustivă constă în a încerca toate cele 2^{56} chei posibile. Această operaţie nu necesită memorie, dar sunt necesare în medie 2^{55} chei pentru a o găsi pe cea bună.

Sau, fiind dat un text clar α, $Oscar$ poate forma anterior (în faza de precalcul) o tabelă cu 2^{56} perechi (β_k, K), ordonată după $\beta_K = e_K(\alpha)$. Atunci când $Oscar$ obţine mesajul β (criptat din textul clar α), el va căuta în tabelă şi va afla imediat cheia K. Astfel, aflarea cheii va necesita un timp de calcul neglijabil (complexitate logaritmică), dar un spaţiu de

memorie gigantic şi un timp de precalcul important. Această variantă nu aduce nici un avantaj din punct de vedere al timpului total, pentru aflarea unei singure chei. Avantajul apare atunci când este necesară căutarea mai multor chei, deoarece tabela precalculată a fost construită o singură dată.

Compromisul spaţiu - timp permite obţinerea unui timp de calcul (precalculul nu se include) inferior celui unei căutări exhaustive, cu un spaţiu de memorie inferior celui necesar reţinerii tuturor cheilor. Algoritmul foloseşte o *funcţie de reducere* R, care micşorează o secvenţă de 64 biţi la una de 56 biţi (de exemplu, R poate şterge pur şi simplu 8 biţi din secvenţa iniţială).

Fie α un text clar de 64 biţi; se defineşte $g(K_0) = R(e_{K_0}(\alpha))$ pentru orice secvenţă K_0 de 56 biţi (rezultatul $g(K_0)$ este de asemenea de lungime 56).

Algoritmul mai foloseşte doi parametri întregi pozitivi m, t. În faza de precalcul, *Oscar* defineşte m secvenţe arbitrare de 56 biţi fiecare, notate $X(i, 0),\ 1 \le i \le m$. Apoi, folosind relaţia de recurenţă

$$X(i, j) = g(X(i, j-1)), \qquad 1 \le i \le m,\ 1 \le j \le t,$$

Oscar determină valorile $X(i, j),\ 1 \le j \le t$, formând cu ele o matrice $X_{m \times t}$.

Din aceste valori, *Oscar* păstrează într-o tabelă T numai $2m$ perechi

$$(X(i, 0), X(i, t)), \qquad 1 \le i \le m$$

(deci sunt memorate numai prima şi ultima coloană a matricii X).

În momentul atacului, *Oscar* obţine textul criptat β al textului clar α ales de el (reamintim, este vorba de un atac cu text clar ales). El va căuta cheia K în cele t coloane ale matricii X, consultând tabela T.

Să presupunem $K = X(i, t-j)$ pentru un j $(1 \le j \le t)$ dat (K este în una din cele t coloane ale lui X). Vom avea $g^j(K) = X(i, t)$, şi

$$g^j(K) = g^{j-1}(g(K)) = g^{j-1}(R(e_K(\alpha))) = g^{j-1}(R(\beta)).$$

Să calculăm şirul β_j $(1 \le j \le t)$ definit prin relaţia de recurenţă

$$\beta_j = \begin{cases} R(\beta) & \text{dacă } j = 1 \\ g(\beta_{j-1}) & \text{dacă } 2 \le j \le t \end{cases}$$

Dacă $K = X(i, t-j)$, vom avea $\beta_j = X(i, t)$.

De remarcat că reciproca nu este adevărată: nu este suficient ca $\beta_j = X(i, t)$ pentru a avea $K = X(i, t-j)$, deoarece funcţia de reducere R nu este injectivă (R reduce un spaţiu de 2^{64} valori în unul de 2^{56} valori, deci fiecare valoare provine în general din $2^8 = 256$ elemente). Trebuie verificată egalitatea $\beta = e_{X(i,t-j)}(\alpha)$, pentru a decide dacă $X(i, t-j)$ este într-adevăr cheia.

Valoarea $X(i, t-j)$ nu este disponibilă în memorie, dar ea se poate recalcula în $t-j$ paşi, plecând de la $X(i, 0)$.

Oscar va folosi deci următorul algoritm:

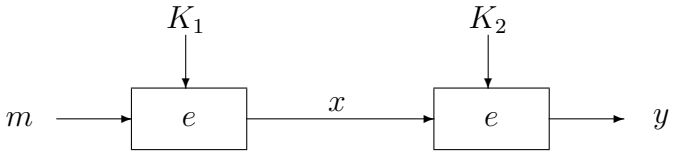
1. $\beta_1 \leftarrow R(\beta)$
2. **for** $j \longleftarrow 1$ **to** t **do**
   2.1. **if** $\exists\, i$ cu $\beta_j = X(i,t)$ **then**
      2.1.1. calculează $X(i,t-j) = g^{t-j}(X(i,0))$
      2.1.2. **if** $\beta = e_{X(i,t-j)}(\alpha)$ **then** $K \leftarrow X(i,t-j)$, $STOP$
   2.2. $\beta_{j+1} \leftarrow g(\beta_j)$

Analizând probabilitatea de succes a algoritmului, se poate arăta că dacă $mt^2 \approx N = 2^{56}$, atunci probabilitatea să găsim cheia K în cele t coloane ale matricii X este de circa $0,8mt/N$. Coeficientul $0,8$ provine din faptul că valorile $X(i,t)$ pot să nu fie distincte. O sugestie este de a alege $m \approx t \approx N^{1/3}$ şi de a construi $N^{1/3}$ tabele, fiecare cu altă funcţie de reducere R. Astfel, spaţiul de memorie necesar este de $112 \cdot N^{2/3}$ biţi (trebuie păstraţi $2N^{2/3}$ valori de 56 biţi). Timpul de precalcul este liniar $\mathcal{O}(N)$.

Timpul de calcul al atacului este mai dificil de evaluat. De remarcat că pasul 2.1 poate fi implementat în cel mai rău caz într-un timp $\mathcal{O}(log\ m)$, utilizând arbori binari de căutare. Dacă acest pas eşuează (deci nu se găseşte nici o valoare), timpul de calcul este $\mathcal{O}(N^{2/3})$. Ceilalţi paşi care urmează cresc acest timp doar cu un factor constant.

6.3 Atacul *meet-in-the-middle*

Este un atac cu text clar ales, dezvoltat în 1981 de Merkle şi Hellman ca răspuns la ideea unei duble criptări cu două chei diferite, conform schemei



Lema 6.1. *Pentru un sistem de criptare bloc, o dublă criptare poate fi atacată folosind* $\mathcal{O}(2^n)$ *operaţii şi* $\mathcal{O}(2^n)$ *spaţiu de memorie, unde n este lungimea cheii.*

Demonstraţie: Fie (α, β) o pereche *(text clar, text criptat)* obţinută pe baza schemei de mai sus.

1. Pentru fiecare din cele 2^n chei posibile se calculează $x_i = e_{K_i}(\alpha)$;
2. Se stochează (x_i, K_i) sortate (sau indexate) după x_i.
3. Pentru fiecare din cele 2^n chei posibile:
   (a) Se calculează $x_j = d_{K_j}(\beta)$;
   (b) Se caută in lista creată la pasul anterior o pereche (x_i, K_i) cu $x_i = x_j$;

4. O pereche de chei posibile este (K_i, K_j);
   (deoarece $e_{K_i}(\alpha) = d_{K_j}(\beta)$, deci $e_{K_j}(e_{K_i}(\alpha)) = \beta$).

Algoritmul se reia cu alte perechi (α, β), până ce perechea de chei folosite este determinată în mod unic.

Exemplul 6.1. *Un atac direct asupra unei duble criptări cu DES ar necesita un timp de ordin* 2^{112} *şi un spaţiu neglijabil. Dacă se foloseşte un atac meet-in-the-middle, timpul va fi* 2^{56} *iar spaţiul* 2^{56}*. Strategii complementare de genul compromisului spaţiu - timp pot duce la variante de genul:* 2^p *spaţiu,* 2^q *timp, unde* $p + q = 112$ *([41]).*

6.4 Criptanaliza diferenţială

Unul din cele mai cunoscute atacuri ale *DES*-ului este *criptanaliza diferenţială*, introdusă de Biham şi Shamir în 1991 ([4],[5],[6]). Cu toate că nu dă o modalitate practică de spargere a funcţiei *DES* în 16 tururi, ea furnizează atacuri eficace pentru variantele de *DES* cu un număr redus de runde. De exemplu, un *DES* cu 8 runde poate fi spart în câteva minute cu un *PC* obişnuit. Ulterior, criptanaliza diferenţială a fost extinsă ca metodă standard de atac pentru toate sistemele de criptare bloc.

6.4.1 Privire generală

Un atac prin criptanaliză diferenţială exploatează faptul că anumite diferenţe dintre texte clare să genereze – cu probabilitate semnificativă – diferenţe fixate între mesajele criptate corespunzătoare.

Astfel, să considerăm un sistem de criptare pe blocuri de n biţi, cu intrarea (textul clar) $\alpha = (a_1, a_2, \ldots, a_n)$ şi ieşirea (textul criptat) $\beta = (b_1, b_2, \ldots, b_n)$.

Fie α' şi α'' două blocuri de intrare, iar β', respectiv β'' blocurile de ieşire corespunzătoare. Diferenţa dintre biţii de intrare este dată de

$$\Delta\alpha = \alpha' \oplus \alpha'' = (\Delta a_1, \Delta a_2, \ldots \Delta a_n)$$

unde $\Delta a_i = a_i' \oplus a_i''$, iar a_i' şi a_i'' reprezintă al i-lea bit din α', respectiv α''.

Similar,

$$\Delta\beta = \beta' \oplus \beta'' = (\Delta b_1, \Delta b_2, \ldots, \Delta b_n)$$

este diferenţa dintre mesajele de ieşire (unde $\Delta b_i = b_i' \oplus b_i''$).

Într-un sistem de criptare ideal, probabilitatea ca, fiind dată o diferenţă de intrare $\Delta\alpha$, să apară o anumită diferenţă de ieşire $\Delta\beta$, este $1/2^n$ (unde n este mărimea blocului de criptare).

Criptanaliza diferenţială exploatează situaţia când pentru o anumită diferenţă de intrare $\Delta\alpha$, o diferenţă de ieşire $\Delta\beta$ apare cu o probabilitate p_D semnificativ mai mare decât $1/2^n$. Perechea $(\Delta\alpha, \Delta\beta)$ pentru care există o astfel de situaţie se numeşte *diferenţială*.

În esenţă, criptanaliza diferenţială este un atac cu text clar ales: *Oscar* alege perechi de intrări (α', α''), care verifică o anumită diferenţă $\Delta\alpha$, ştiind că pentru această valoare $\Delta\alpha$, există o diferenţială $(\Delta\alpha, \Delta\beta)$.

Cel mai important pas constă în construirea unei diferenţiale $(\Delta\alpha, \Delta\beta)$, folosind biţii textului clar pentru α, şi intrarea în ultima rundă a sistemului de criptare pentru β.

Aceasta se poate realiza examinând diverse *diferenţiale caracteristice*. O *diferenţială caracteristică* este o secvenţă $(\Delta\alpha_1, \Delta\alpha_2, \ldots, \Delta\alpha_n)$ unde $(\Delta\alpha_i, \Delta\alpha_{i+1})$ este o diferenţială a rundei i din sistemul de criptare (deci $\alpha_1 = \alpha,\ \alpha_n = \beta$).

Pe baza unei diferenţiale caracteristice putem exploata informaţia care intră în ultima rundă a sistemului, pentru a afla o parte din biţii cheii.

Deci, o primă problemă constă în construirea de diferenţiale caracteristice. Pentru aceasta vom examina şi exploata diverse particularităţi ale S - boxurilor.

Practic, vom considera toate diferenţele de intrare şi ieşire din fiecare S-box, determinând perechile $(\Delta\alpha, \Delta\beta)$ cu probabilităţi mari. Combinând aceste perechi de la o rundă la alta, astfel ca diferenţele dintre două ieşiri dintr-o rundă să coincidă cu diferenţele dintre două intrări în runda următoare, vom determina o diferenţială între două texte clare şi intrările a două texte în ultima rundă. În final, biţii ultimei sub-chei vor dispare (deoarece sunt folosiţi în ambele texte şi – prin diferenţiere cu operaţia XOR – se anulează reciproc).

6.4.2 Analiza componentelor unui sistem de criptare

Să examinăm perechile de diferenţe ale unui S-box.
Vom considera un 4×4 S-box, cu intrarea $\alpha = (a_1, a_2, a_3, a_4)$ şi ieşirea $\beta = (b_1, b_2, b_3, b_4)$.

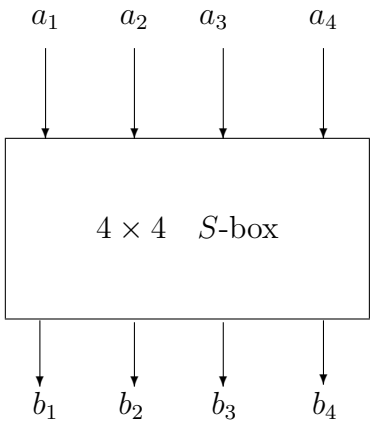


Figura 1: Modelul unui 4×4 S - box

Toate perechile de diferenţe $(\Delta\alpha, \Delta\beta)$ ale unui astfel de S - box vor fi examinate; fixăm o diferenţă $\Delta\alpha$ şi calculăm probabilităţile de apariţie pentru toate diferenţele de ieşiri $\Delta\beta$, peste mulţimea perechilor de intrare (α', α'') cu $\alpha' \oplus \alpha'' = \Delta\alpha$.

Deoarece ordinea perechilor nu este relevantă, pentru un 4×4 S-box sunt necesare doar cele 16 valori posibile ale unei intrări α'; a doua intrare α'' va fi calculată cu formula $\alpha'' = \alpha' \oplus \Delta\alpha$.

Toată construcţia acestei secţiuni va fi exemplificată folosind prima linie a primului S-box din DES (am considerat cel mai semnificativ bit din notaţia hexazecimală ca fiind cel mai din stânga bit al S-boxului):

Intrare	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*	*D*	*E*	*F*
Iesire	*E*	4	*D*	1	2	*F*	*B*	8	3	*A*	6	*C*	5	9	0	7

Tabela 1. Reprezentarea (in hexazecimal) a unui S - box

Fiecare pereche $(intrare, iesire)$ poate fi considerată ca un 4×4 S-box cu intrări şi ieşiri pe 4 biţi.

Putem calcula diferenţa de ieşire $\Delta\beta$ pentru orice pereche de intrare $(\alpha', \alpha'' = \alpha' \oplus \Delta\alpha)$.

De exemplu, Tabela 2 prezintă valorile (binare) pentru α, β, şi valorile $\Delta\beta$ pentru perechile de intrare $(\alpha, \alpha \oplus \Delta\alpha)$, unde $\Delta\alpha$ are valorile

1011 (hex B), 1000 (hex 8) şi 0100 (hex 4).

α	β	$\Delta\alpha = 1011$	$\Delta\alpha = 1000$	$\Delta\alpha = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	0101	0110
0011	0001	0010	1011	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	0101	0110
1011	1100	0010	1011	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011

Tabela 2. Perechi de diferenţe într-un S - box

De exemplu, pentru $\alpha = 0000$ şi $\Delta\alpha = 1011$ avem $\alpha' = \alpha \oplus \Delta\alpha = 1011$. Ieşirea (în binar) pentru intrarea α' este, conform Tabelei 1, $\beta' = 1100$ (hex C), deci $\Delta\beta = \beta \oplus \beta' = 1110 \oplus 1100 = 0010$, valoare care apare pe coloana a treia din tabelă.

Analizând Tabela 2, vedem că numărul apariţiilor lui $\Delta\beta = 0010$ pentru $\Delta\alpha = 1011$ (coloana 3) este 8 din 16 valori posibile (deci probabilitate 8/16). Similar, numărul de apariţii $\Delta\beta = 1011$ pentru $\Delta\alpha = 1000$ este 4 (din 16), iar numărul de apariţii ale lui $\Delta\beta = 1010$ pentru $\Delta\alpha = 0100$ este 0 (din 16).

Putem aduna toate datele unui S - box într-o tabelă de distribuţie a diferenţelor, în care liniile reprezintă valorile $\Delta\alpha$, iar coloanele – valorile $\Delta\beta$ (toate în hexazecimal).

Revenind la exemplul nostru, tabela de distribuţie a diferenţelor pentru S-boxul din Tabela 1, este dată de Tabela 3:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Tabela 3. Tabela de distribuţie a diferenţelor

Fiecare element al tabelei reprezintă numărul de apariţii ale diferenţei de ieşire $\Delta\beta$, fiind dată diferenţa de intrare $\Delta\alpha$.

Observaţia 6.1.

1. *Înafară de situaţia specială ($\Delta\alpha = 0,\ \Delta\beta = 0$), cea mai mare valoare din tabelă este 8; ea corespunde lui $\Delta\alpha = B$ şi $\Delta\beta = 2$. Deci, pentru o pereche de intrări arbitrare cu $\Delta\alpha = B$, probabilitatea ca $\Delta\beta = 2$ este $8/16 = 1/2$.*

2. *Cea mai mică valoare din tabelă este 0; în acest caz, probabilitatea de apariţie a valorii $\Delta\beta$ pentru $\Delta\alpha$ corespunzător, este 0.*

3. *Suma elementelor pe fiecare linie este $2^n = 16$; similar, suma pe fiecare coloană este tot $2^n = 16$.*

4. *Toate valorile din tabel sunt numere pozitive pare; motivul este acela că o pereche de valori de intrare/ieşire* (α', α'') *are aceeaşi diferenţă* $\Delta\alpha$ *ca şi perechea* (α'', α').

5. *Pentru un* S *- box injectiv, o diferenţă de intrare* $\Delta\alpha = 0$ *va duce la o diferenţă de ieşire* $\Delta\beta = 0$.

   Ca o consecinţă, prima linie şi prima coloană vor fi 0*, cu excepţia intersecţiei lor, unde va fi valoarea* $2^n = 16$.

Dacă s-ar putea construi un S - box ideal, care să nu ofere nici o informaţie de tip diferenţial între intrări şi ieşiri, atunci toate valorile din tabela sa de distribuţie a diferenţelor vor fi egale cu 1, iar probabilitatea de apariţie a unei diferenţe $\Delta\beta$ atunci când diferenţa de intrare este $\Delta\alpha$, va fi $1/2^n = 1/16$.

Totuşi pe baza proprietăţilor arătate în Observaţia 6.1, un astfel de S - box ideal nu este posibil.

6.4.3 Reţea substituţie - permutare

O *reţea substituţie - permutare* (*Substitution Permutation Network* – SPN) este un sistem de criptare bloc, format din mai multe runde, fiecare rundă conţinând o substituţie şi o permutare. Cheia este expandată în mod similar sistemelor uzuale şi fiecare cheie de rundă va fi XOR-ată cu mesajul la începutul fiecărei runde.

Un SPN păstrează proprietăţile generale folosite în construirea sistemelor de criptare uzuale (inclusiv DES şi AES).

În prezentarea criptanalizei diferenţiale (şi apoi a celei liniare) vom utiliza un SPN care criptează un text clar de 16 biţi într-un mesaj de 16 biţi. Cheia are de asemenea 16 biţi şi este expandată în 5 sub-chei de rundă, de câte 16 biţi fiecare. La fiecare rundă, cei 16 biţi de intrare sunt XOR - aţi cu cheia de rundă, apoi sunt separaţi în patru blocuri de câte 4 biţi, iar fiecare bloc trece prin câte un S - box. La sfârşitul rundei, cei 16 biţi sunt permutaţi.

În cazul nostru, vom folosi acelaşi S - box pentru toate blocurile şi toate rundele; anume S - boxul definit de Tabela 1 (prima linie din S_1 - boxul definit în DES).

Permutarea utilizată este aceiaşi pentru toate rundele:

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

Tabela 4: O permutare utilizată în construirea unui SPN

Figura 2 prezintă schematic structura unui SPN:

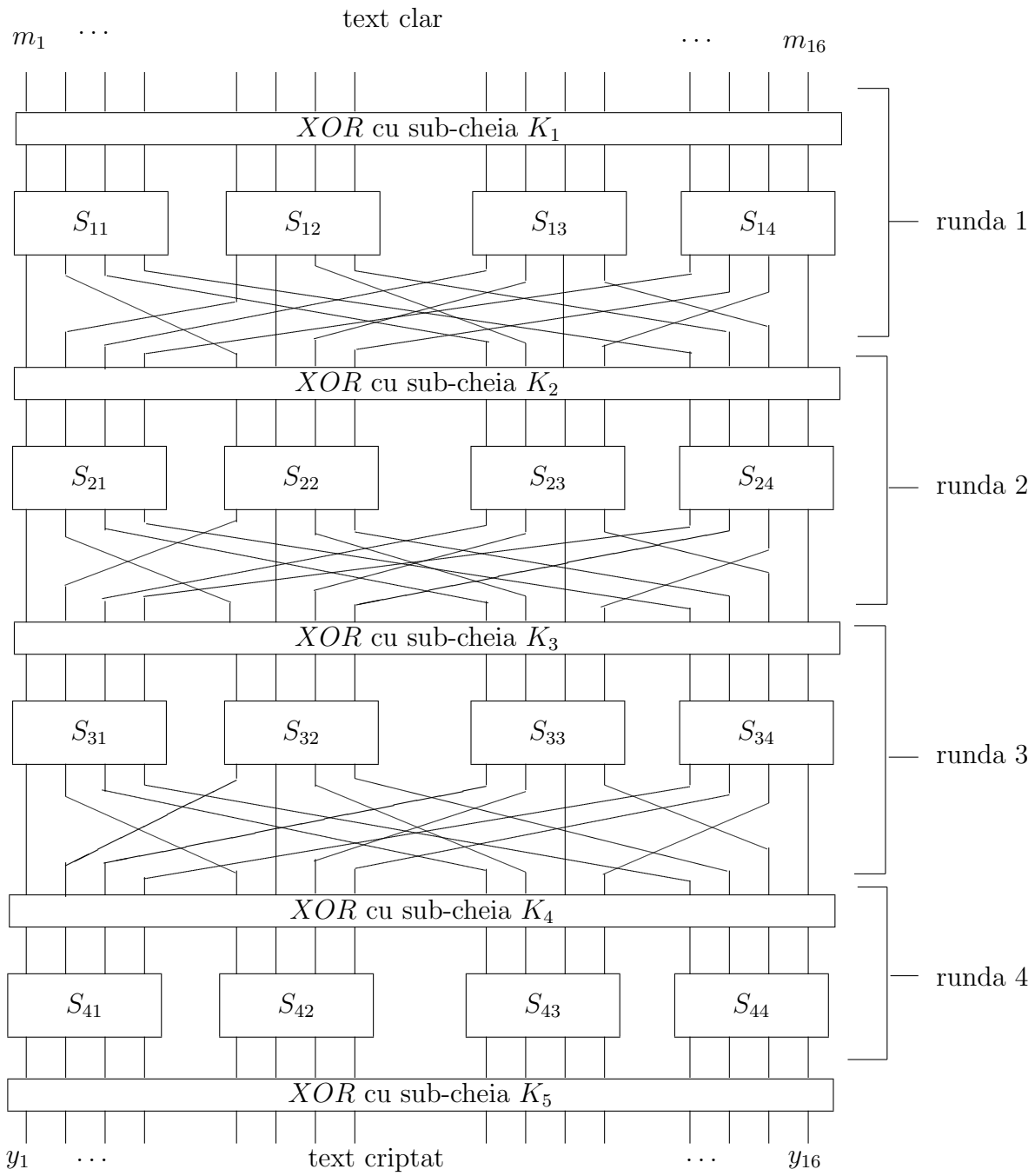


Figura 2: Structura generală a unui SPN.

6.4.4 Diferenţiala caracteristică într-un SPN

În această secţiune vom construi o modalitate de atac prin criptanaliză diferenţială asupra unui SPN, având ca scop aflarea unui set de biţi din sub-cheia folosită în ultima rundă.

Să considerăm SPN-ul din Figura 2.

Vom nota cu U_i intrarea în S - boxurile din runda i, iar cu V_i – ieşirea din S - boxuri în runda i.

Să începem cu două texte clare, a căror diferenţă este

$$\Delta U_1 = [0000\ 1011\ 0000\ 0000]$$

Deci singurul S - box activ în prima rundă este S_{12} (celelalte S - boxuri sunt inactive: intrarea este 0, deci şi ieşirea va fi tot 0). Folosind Tabela 2, obţinem diferenţa de ieşire

$$\Delta V_1 = [0000\ 0010\ 0000\ 0000]$$

Altfel spus,

$S_{12}: \quad \Delta\alpha = B_{16} = 1011 \quad \Longrightarrow \quad \Delta\beta = 2_{16} = 0010$ cu probabilitate $8/16 = 1/2$.

Urmărind acum permutarea de la sfârşitul primei runde, găsim mesajul de intrare în S - boxurile celei de-a doua runde:

$$\Delta U_2 = [0000\ 0000\ 0100\ 0000]$$

cu probabilitate 1/2.

În a doua rundă, singurul S - box activ este S_{23}. Vom avea perechea de diferenţe

$S_{23}: \quad \Delta\alpha = 4_{16} = 0100 \quad \Longrightarrow \quad \Delta\beta = 6_{16} = 0110$ cu probabilitate $6/16 = 3/8$.

Am obţinut deci ieşirea

$$\Delta V_2 = [0000\ 0000\ 0110\ 0000]$$

iar permutarea de la sfârşitul rundei a doua va da

$$\Delta U_3 = [0000\ 0010\ 0010\ 0000]$$

cu probabilitate 3/8.

În a treia rundă sunt active două S - boxuri: S_{32} şi S_{33}:

$S_{32}: \quad \Delta\alpha = 2_{16} = 0010 \quad \Longrightarrow \quad \Delta\beta = 5_{16} = 0101$ cu probabilitate $6/16 = 3/8$,

$S_{33}: \quad \Delta\alpha = 2_{16} = 0010 \quad \Longrightarrow \quad \Delta\beta = 5_{16} = 0101$ cu probabilitate $6/16 = 3/8$.

Deci, ieşirea din S - boxurile din runda a treia este

$$\Delta V_3 = [0000\ 0101\ 0101\ 0000]$$

iar intrarea în runda a patra (după efectuarea permutării) este

$$\Delta U_4 = [0000\ 0110\ 0000\ 0110]$$

Cu alte cuvinte,

$$\Delta U_1 = [0000\ 1011\ 0000\ 0000] \quad \Longrightarrow \quad \Delta V_3 = [0000\ 0101\ 0101\ 0000]$$

şi

$$\Delta V_3 = [0000\ 0101\ 0101\ 0000] \quad \Longrightarrow \quad \Delta U_4 = [0000\ 0110\ 0000\ 0110]$$

Rezultă

$$\Delta U_1 = [0000\ 1011\ 0000\ 0000] \quad \Longrightarrow \quad \Delta U_4 = [0000\ 0110\ 0000\ 0110]$$

Tabela 3 de distribuţie a diferenţelor poate conduce la ideea unor probabilităţi condiţionale independente (noţiune fără utilizare în matematică, dar extrem de intuitivă în această situaţie):

$$Pr[\Delta\beta|\Delta\alpha] = (\Delta\alpha, \Delta\beta)$$

Pe baza acestei relaţii, diferenţialele pot fi compuse. Astfel, pentru cazul studiat mai sus, vom avea

$$(0000\ 1011\ 0000\ 0000,\ 0000\ 0110\ 0000\ 0110) = \frac{1}{2} \cdot \frac{3}{8} \cdot \frac{3}{8} \cdot \frac{3}{8} = \frac{27}{1024}$$

Altfel spus, diferenţiala caracteristică (0000 1011 0000 0000, 0000 0110 0000 0110) apare cu probabilitatea 27/1024.

Să vedem cum pot fi extraşi acum biţii din cheie, pe baza acestei informaţii.

6.4.5 Extragerea biţilor din cheie

După ce am obţinut diferenţiala caracteristică a rundelor, putem construi un atac din care să rezulte o parte din biţii care formează cheia de criptare.

Într-un sistem de criptare bloc cu N runde, vom ataca runda $N-1$. Astfel, în cazul reţelei SPN, vom putea extrage biţi din sub-cheia K_5 atacând a patra rundă.

Pentru o decriptare a textului obţinut după ultima rundă, trebuie să mergem în sens invers: anume, realizăm un XOR între textul criptat şi sub-cheia ultimei runde – care este influenţată de diferenţialele nenule – iar apoi trecem datele în sens invers prin S-boxuri. Operaţia XOR o realizăm folosind toate valorile posibile ale sub-cheii.

Rezumând, vom face un XOR între textul criptat cu toate sub-cheile posibile, după care vom aplica inversele S-boxelor.

Această căutare a cheii poate fi mult redusă dacă luăm numai perechile "bune" (perechile pentru care se obţin diferenţiale caracteristice).

De exemplu, o pereche "bună" a fost obţinută în secţiunea precedentă: ea are $U_{41} = U_{43} = 0000$ iar U_{42} şi U_{44} conţin diferenţiale nenule (U_{ij} reprezintă intrarea în S - boxul j din runda i).

Decriptarea este obţinută pentru toate perechile de mesaje criptate care corespund perechilor de texte clare având diferenţa de intrare fixată $\Delta\alpha$.

Figura 3 trasează modalitatea de decriptare pentru diferenţiala caracteristică detaliată în secţiunea precedentă.

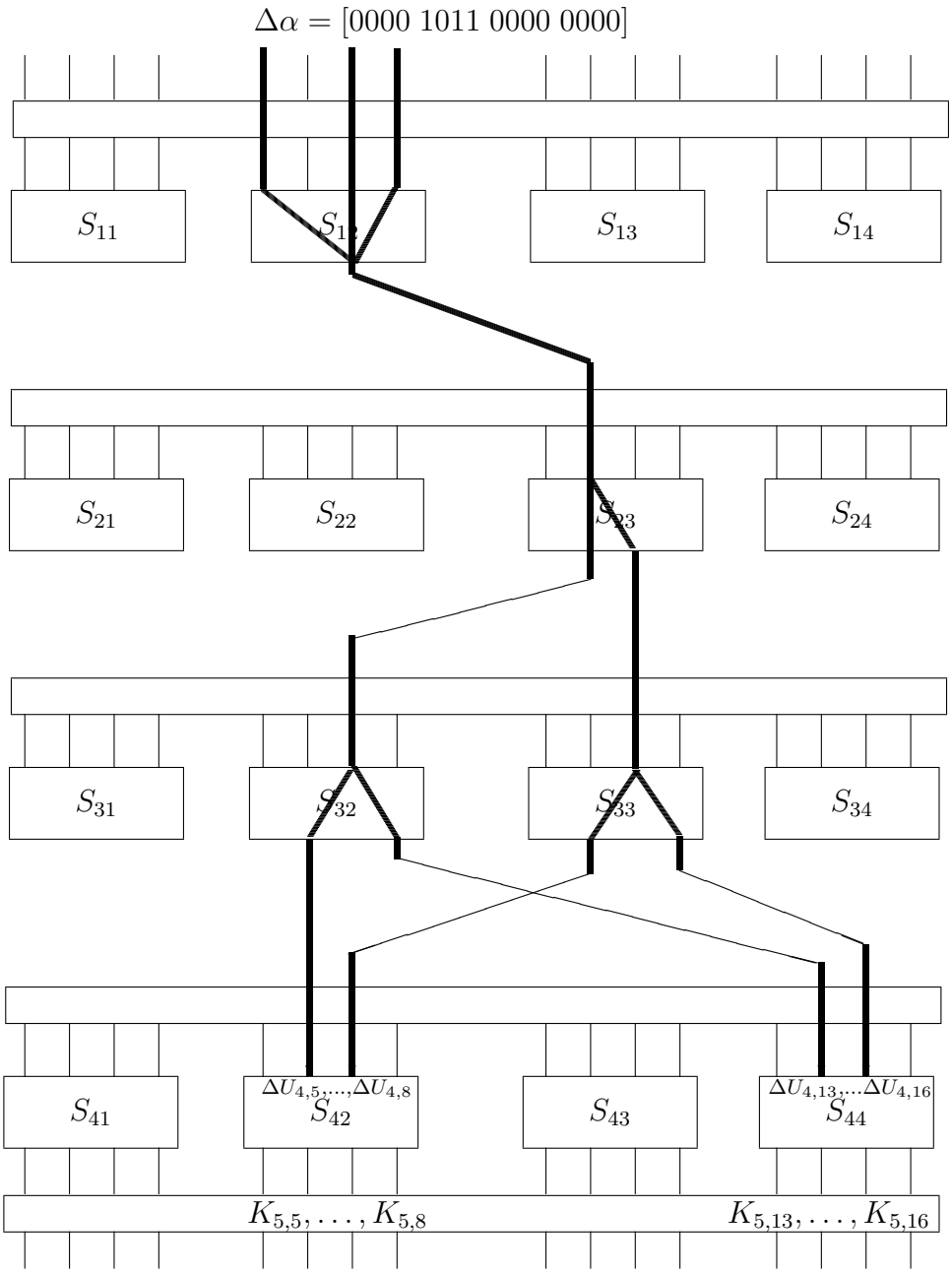


Figura 3: Diferenţiala caracteristică, cu $\Delta\alpha$ ca diferenţă de intrare

Pentru fiecare sub-cheie posibilă, ai cărei biţi sunt influenţaţi de diferenţele nenule din diferenţiale, definim câte un counter. Aceşti counteri sunt incrementaţi atunci când intrarea în ultima rundă coincide cu valoarea prevăzută de diferenţiala caracteristică.

În final, sub-cheia cu valoare maximă a counterului ataşat va fi considerată drept cea corectă; restul biţilor din sub-cheie vor fi găsiţi cu o căutare prin forţă brută.

În exemplul din Figura 2, diferenţiala caracteristică a influenţat S - boxurile S_{42} şi S_{44} din ultima rundă; deci – pentru orice pereche de mesaje criptate – încercăm toate cele 256 variante pentru $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$, şi – de câte ori diferenţa la intrarea în ultima rundă determinată de decriptare este

$$\Delta U_4 = [0000\ 0110\ 0000\ 0110]$$

va fi incrementat counterul asociat.

6.4.6 Criptanaliza diferenţială pentru DES

Într-un astfel de atac se ignoră permutarea iniţială IP şi inversa sa (ele nu joacă nici un rol în criptanaliză). Ne vom mărgini la un DES restrâns pe n ($n \leq 16$) runde.

Deci textul clar este L_0R_0, iar cel criptat – L_nR_n (vom ignora de asemenea inversarea finală dintre L_n şi R_n).

Pentru criptanaliza diferenţială vom considera două texte clare L_0R_0, $L_0^*R_0^*$ şi textele criptate corespunzătoare L_nR_n respectiv $L_n^*R_n^*$.
Diferenţele sunt $L_0'R_0' = L_0R_0 \oplus L_0^*R_0^*$ şi $L_n'R_n' = L_nR_n \oplus L_n^*R_n^*$.

Definiţia 6.1. *Fie S_j ($1 \leq j \leq 8$) un S - box din sistemul de criptare DES. Pentru orice pereche (B_j, B_j^*) de şiruri de 6 biţi, vom defini XOR-ul de intrare pentru S_j prin $B_j' = B_j \oplus B_j^*$, iar XOR-ul de ieşire prin $S_j(B_j) \oplus S_j(B_j^*)$.*

De remarcat că un XOR de intrare este o secvenţă de 6 biţi, iar un XOR de ieşire este o secvenţă de 4 biţi.

Definiţia 6.2. *Pentru orice $B_j' \in Z_2^6$ se notează $\Delta(B_j')$ mulţimea perechilor (B_j, B_j^*) care prin XOR dau B_j'.*

Observaţia 6.2.

- *O mulţime $\Delta(B_j')$ conţine $2^6 = 64$ elemente;*
- $\Delta(B_j') = \{(B_j, B_j \oplus B_j') |\ B_j \in Z_2^6\}$.

Pentru fiecare pereche din $\Delta(B_j')$ vom calcula XOR-ul de ieşire al lui S_j şi construim o tabelă de distribuţii ale valorilor obţinute (sunt 64 ieşiri pe un spaţiu de $2^4 = 16$ valori posibile). Pe această tabelă se va baza atacul de criptanaliză diferenţială.

Exemplul 6.2. *Să considerăm primul S - box S_1 şi XOR-ul de intrare* 110100*. Vom avea:*

$$\Delta(110100) = \{(000000, 110100), (000001, 110101), \ldots, (111111, 001011)\}.$$

Pentru fiecare pereche din $\Delta(110100)$ *vom calcula XOR-ul de ieşire al lui S_1. De exemplu,* $S_1(000000) = E_{16} = 1110$, $S_1(110100) = 9_{16} = 1001$ *deci XOR-ul de ieşire S_1 al perechii* $(000000, 110100)$ *este* 0111*.*

Efectuând acest calcul pentru toate cele 64 *perechi din* $\Delta(110100)$*, vom obţine distribuţia următoare a XOR-urilor de ieşire pentru S_1:*

0000	0001	0010	0011	0100	0101	0110	0111
0	8	16	6	2	0	0	12
1000	1001	1010	1011	1100	1101	1110	1111
6	0	0	0	0	8	0	6

În Exemplul 6.2 au apărut numai 8 din cele 16 valori de ieşire posibile. În general, dacă se fixează un S - box S_j şi un XOR de intrare diferit de 000000, se constată că vor apare aproximativ $75 - 80$ % din valorile posibile de ieşire.

Definiţia 6.3. *Pentru* $1 \leq j \leq 8$ *şi secvenţele B_j', C_j' de* 6 *respectiv* 4 *biţi, definim*

$$IN_j(B_j', C_j') = \{B_j \in Z_2^6 |\ S_j(B_j) \oplus S_j(B_j \oplus B_j') = C_j',$$

$$N_j(B_j', C_j') = card(IN_j(B_j', C_j')).$$

Distribuţia dată în Exemplul 6.2 dă valorile $N_1(110100, C_1')$, $C_1' \in Z_2^4$. Toate aceste valori se găsesc în Tabela 5.

Pentru fiecare din cele 8 S - boxuri există 64 XOR-uri de intrare posibile; deci în total vor fi 512 date de distribuit, lucru uşor de realizat cu un calculator.

Reamintim că intrarea într-un S - box la runda i este $B = E \oplus J$, unde $E = E(R_{i-1})$ este rezultatul expandării lui R_{i-1}, iar $J = K_i$ este un subşir reordonat al cheii K. XOR-ul de intrare (al celor 8 S - boxuri) este deci

$$B \oplus B^* = (E \oplus J) \oplus (E^* \oplus J) = E \oplus E^*.$$

De remarcat că XOR-urile de intrare nu depind de sub-cheia J, pe când XOR-urile de ieşire depind.

XOR de ieşire	intrări posibile
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 5: Distribuţia valorilor intrare/ieşire pentru S_1

Să detaliem pe grupuri de câte 6 biţi, cuvintele cu care se lucrează:

$$B = B_1B_2B_3B_4B_5B_6B_7B_8, \quad E = E_1E_2E_3E_4E_5E_6E_7E_8, \quad J = J_1J_2J_3J_4J_5J_6J_7J_8$$

În mod similar se scriu B^* şi E^*. Să presupunem acum că se ştiu valorile E_j şi E_j^* pentru un j $(1 \leq j \leq 8)$ dat, precum şi valoarea XOR de ieşire $C_j' = S_j(B_j) \oplus S_j(B_j^*)$ a lui S_j. Vom avea

$$E_j \oplus J_j \in IN_j(E_j', C_j')$$

unde $E_j' = E_j \oplus E_j^*$.

Fie mulţimea $test_j$ definită astfel:

$$test_j(E_j, E_j^*, C_j') = \{B_j \oplus E_j |\ B_j \in IN_j(E_j', C_j')\}$$

(s-au luat toate XOR-urile lui E_j cu elemente din $IN_j(E_j', C_j')$).

Din aceste consideraţii rezultă imediat teorema:

Teorema 6.1. *Dacă E_j, E_j^* sunt subsecvenţe construite pentru intrarea în S - boxul S_j, iar C_j' este XOR-ul de ieşire al lui S_j, atunci biţii sub-cheii J_j apar în $test_j(E_j, E_j^*, C_j')$.*

Cum se poate remarca, există exact $N_j(E_j', C_j')$ secvenţe de 6 biţi în $test_j(E_j, E_j^*, C_j')$; valoarea corectă J_j este una din acestea.

Exemplul 6.3. *Să considerăm* $E_1 = 000001$, $E_1^* = 110101$, $C_1{}' = 1101$. *Deoarece* $N_1(110100, 1101) = \{000110, 010000, 010110, 011100, 100010, 101000, 110010\}$ *are* 8 *elemente, există* 8 *secvenţe posibile pentru* J_1, *cumulate în* $test_1(000001, 110101, 1101) = \{000111, 010001, 010111, 011101, 100011, 100101, 101001, 110011\}$.

Dacă se ia un alt triplet $(E_1, E_1^*, C_1{}')$, *vom obţine altă mulţime* $test_1$ *cu valori pentru* J_1, *deci valoarea corectă se va găsi în intersecţia lor.*

Atacul unui DES definit pe trei runde

Să vedem cum se aplică aceste idei pentru un DES construit pe 3 runde. Începem cu o pereche de texte clare L_0R_0 şi $L_0^*R_0^*$, criptate în L_3R_3 respectiv $L_3^*R_3^*$. Vom avea

$$R_3 = L_2 \oplus f(R_2, K_3) = R_1 \oplus f(R_2, K_3) = L_0 \oplus f(R_0, K_1) \oplus f(R_2, K_3).$$

Construcţia pentru R_3^* este similară. Deci

$$R_3{}' = L_0{}' \oplus f(R_0, K_1) \oplus f(R_0^*, K_1) \oplus f(R_2, K_3) \oplus f(R_2^*, K_3).$$

Să presupunem că s-a ales $R_0 = R_0^*$, deci $R_0{}' = 00\ldots0$.

Atunci $f(R_0, K_1) = f(R_0^*, K_1)$ şi deci $R_3{}' = L_0{}' \oplus f(R_2, K_3) \oplus f(R_2^*, K_3)$.

$R_3{}'$ se poate calcula pe baza textelor criptate, iar $L_0{}'$ – pe baza textelor clare; deci se poate determina $f(R_2, K_3) \oplus f(R_2^*, K_3) = R_3{}' \oplus L_0{}'$.

Avem $f(R_2, K_3) = P(C)$, $f(R_2^*, K_3) = P(C^*)$ unde C respectiv C^* sunt ieşirile corespunzătoare din cele 8 S - boxuri (reamintim, P este o permutare fixată din descrierea sistemului DES). Deci $P(C) \oplus P(C^*) = R_3{}' \oplus L_0$, de unde

$$C' = C \oplus C^* = P^{-1}(R_3{}' \oplus L_0{}').$$

Acesta este XOR-ul de ieşire din cele opt S - boxuri, după a treia rundă.

$R_2 = L_3$ şi $R_2^* = L_3^*$ sunt cunoscute (componente ale textelor criptate); deci se poate calcula

$$E = E(L_3), \quad E^* = E(L_3^*)$$

folosind funcţia de expansiune E. Aceste valori sunt intrările în S - boxuri la runda a treia. Se cunosc deci E, E^*, C' la a treia rundă şi se poate trece – aşa cum am văzut – la construcţia mulţimilor $test_1, test_2, \ldots, test_8$ de valori posibile pentru $J_1, J_2, \ldots, J_8$.

Un algoritm pentru această metodă este formalizat mai jos. Atacul foloseşte mai multe triplete E, E^*, C'. Se utilizează opt tabele de valori şi se determină astfel cei 48 biţi ai subcheii K_3 de la a treia rundă. Cheia de 56 biţi se calculează apoi printr-o căutare exhaustivă a celor $2^8 = 256$ posibilităţi pentru cei 8 biţi rămaşi necunoscuţi.

Intrare: L_0R_0, $L_0^*R_0^*$, L_3R_3, $L_3^*R_3^*$ cu $R_0 = R_0^*$.
1. $C' \leftarrow P^{-1}(R_3{}' \oplus L_0{}')$
2. $E \leftarrow E(L_3)$, $E^* \leftarrow E(L_3^*)$
3. **for** $j := 1$ **to** 8 **do** $test_j(E_j, E_j^*, C_j)$.

6.5 Criptanaliza liniară

Criptanaliza liniară este – împreună cu cea diferenţială – unul din atacurile de bază utilizate contra sistemelor de criptare bloc.

Autorul ei este Mitsuru Matsui, care a aplicat-o prima oară în 1992, ca un atac asupra sistemului de criptare FEAL ([34]). Ulterior, Matsui a publicat un atac similar asupra sistemului DES, probabil prima criptanaliză experimentală prezentată public asupra unui sistem de criptare bloc ([36],[35],[37]).

Practic, un atac asupra sistemului DES nu este însă viabil, el necesitând 2^{43} texte clare cunoscute.

Ulterior au fost construite diverse rafinări ale atacurilor prin criptanaliză liniară, folosind aproximări liniare multiple sau aproximări locale cu expresii neliniare.

6.5.1 Modalitatea generală de atac

Criptanaliza liniară este un atac cu text clar ales şi foloseşte o relaţie liniară de aproximare pentru descrierea sistemului de criptare bloc. Fiind dat un număr suficient de mare de perechi *(text clar, text criptat)* se pot obţine informaţii despre cheie, cu o probabilitate acceptabilă.

Vom ilustra acest atac folosind aceiaşi structură SPN definită în Figura 2 pentru criptanaliza difererenţială. El este construit plecând de la prima linie din S_1 - boxul DES (privită ca un 4×4 S - box) şi permutarea definită în Tabela 4.

Ideea de bază este de a aproxima statistic funcţionarea unei porţiuni din sistemul de criptare printr-o expresie liniară (faţă de operaţia XOR). O astfel de expresie este de forma

$$a_{i_1} \oplus a_{i_2} \oplus \ldots \oplus a_{i_u} \oplus b_{j_1} \oplus b_{j_2} \oplus \ldots \oplus b_{j_v} = 0 \qquad (1)$$

unde a_i reprezintă al i-lea bit al intrării $\alpha = (a_1, a_2, \ldots)$, iar b_j reprezintă al j-lea bit al ieşirii $\beta = (b_1, b_2, \ldots)$.

Scopul criptanalizei liniare este de a determina expresii de forma (1), cu o probabilitate mare sau mică de apariţie (deci cât mai diferită de 1/2).

Evident, rezistenţa unui sistem de criptare la criptanaliza liniară este cu atât mai mare cu cât astfel de relaţii sunt mai puţine. Existenţa unei relaţii de tip (1) – cu probabilitate mare sau mică – denotă o abilitate scăzută a sistemului de criptare privind difuzia datelor criptate.

În general, dacă alegem aleator valori pentru $u + v$ biţi şi le plasăm într-o relaţie de tip (1), probabilitatea ca relaţia să fie verificată va fi 1/2. Cu cât o astfel de probabilitate va fi mai depărtată de 1/2, cu atât va fi mai uşor pentru criptanalist să atace sistemul folosind relaţia respectivă.

Dacă o expresie de tip (1) are loc cu probabilitate p_L pentru texte clare şi texte criptate alese aleator, vom nota *tendinţa*[1] relaţiei respective prin $\epsilon_L = p_L - 1/2$.

Cu cât magnitudinea tendinţei $|p_L - 1/2|$ este mai mare, cu atât mai puţine texte clare va avea nevoie un atac ca să reuşească.

Observaţia 6.3. *Dacă $p_L = 1$, atunci expresia liniară* (1) *reprezintă în totalitate comportarea sistemului de criptare; deci acesta nu va asigura nici o securitate.*

Dacă $p_L = 0$, atunci (1) *reprezintă o relaţie afină în sistemul de criptare, de asemenea indiciul unei slăbiciuni catastrofice.*

Dacă operaţia de adunare folosită este mod 2 (deci XOR), atunci o funcţie afină este complementara unei funcţii liniare. Atât aproximările liniare cât şi cele afine, indicate de $p_L > 1/2$ respectiv $p_L < 1/2$, sunt folosite în egală măsură de criptanaliza liniară.

În mod natural apare întrebarea:

Cum pot fi găsite expresii statistic liniare (deci utilizabile în criptanaliza liniară) ?

Evident, problema se pune relativ la componenta neliniară a sistemelor de criptare, deci relativ la S - boxuri. Când evaluăm proprietăţile neliniare ale unui S - box, este posibil să găsim aproximări liniare între seturi de biţi de intrare şi de ieşire din S - box. Ulterior, este posibil să concatenăm astfel de aproximări liniare ale diverselor S - boxuri, astfel încât să eliminăm biţii intermediari (aflaţi în interiorul sistemului de criptare) şi să obţinem o expresie liniară – cu tendinţă semnificativă – între seturi de biţi din textul clar şi biţi de intrare în ultima rundă.

6.5.2 Lema Piling-Up

Fie $X_1,\ X_2$ două variabile aleatoare. Relaţia $X_1 \oplus X_2 = 0$ este o expresie liniară, echivalentă cu $X_1 = X_2$. Similar, $X_1 \oplus X_2 = 1$ este o expresie afină echivalentă cu $X_1 \neq X_2$.

Să considerăm o probabilitate definită

$$Pr[X_1 = i] = \begin{cases} p_1, & \text{dacă } i = 0 \\ 1 - p_1 & \text{dacă } i = 1 \end{cases}$$

şi

$$Pr[X_2 = i] = \begin{cases} p_2, & \text{dacă } i = 0 \\ 1 - p_2 & \text{dacă } i = 1 \end{cases}$$

Dacă cele două variabile aleatoare sunt independente, atunci

$$Pr[X_1 = i,\ X_2 = j] = \begin{cases} p_1 p_2, & \text{dacă } i = 0,\ j = 0 \\ p_1(1 - p_2), & \text{dacă } i = 0,\ j = 1 \\ (1 - p_1)p_2, & \text{dacă } i = 1,\ j = 0 \\ (1 - p_1)(1 - p_2), & \text{dacă } i = 1,\ j = 1 \end{cases}$$

[1] în engleză *bias*.

şi se poate arăta imediat că

$Pr[X_1 \oplus X_2 = 0] = Pr[X_1 = X_2] = Pr[X_1 = 0, X_2 = 0] + Pr[X_1 = 1, X_2 = 1] = p_1p_2 + (1-p_1)(1-p_2)$

Pentru a simetriza formulele, introducem o nouă variabilă numită *tendinţă* definită

$$\epsilon_i = p_i - \frac{1}{2}, \qquad \epsilon_i \in \left[-\frac{1}{2}, \frac{1}{2}\right]$$

Atunci

$$Pr[X_i = 0] = \frac{1}{2} + \epsilon_i, \qquad Pr[X_i = 1] = \frac{1}{2} - \epsilon_i$$

Lema 6.2. *(Piling - up): Fie $X_1, X_2, \ldots X_n$ variabile aleatoare independente şi pentru $1 \le i_1 < i_2 < \ldots < i_k \le n$, fie $\epsilon_{i_1,i_2,\ldots i_k}$ tendinţa variabilei aleatoare $X_{i_1} \oplus X_{i_2} \oplus \ldots \oplus X_{i_k}$. Atunci*

$$\epsilon_{i_1,i_2,\ldots i_k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_{i_j}.$$

sau – echivalent:

$$Pr[X_1 \oplus \ldots \oplus X_k = 0] = 1/2 + 2^{k-1} \prod_{i=1}^{k} \epsilon_i$$

Demonstraţie. Prin inducţie după k. Pentru $k = 1$ egalitatea este banală. Pentru $k = 2$ se verifică uşor relaţia $\epsilon_{i_1,i_2} = 2\epsilon_{i_1}\epsilon_{i_2}$.

Presupunem egalitatea adevărată pentru $k = p$ şi să o arătăm pentru $k = p + 1$. Din faptul că tendinţa lui $X_{i_1} \oplus \ldots \oplus X_{i_p}$ este $2^{p-1} \prod_{j=1}^{p} \epsilon_{i_j}$ rezultă

$$Pr[X_{i_1} \oplus \ldots \oplus X_{i_p} = 0] = \frac{1}{2} + 2^{p-1} \prod_{j=1}^{p} \epsilon_{i_j} \quad \text{şi}$$

$$Pr[X_{i_1} \oplus \ldots \oplus X_{i_p} = 1] = \frac{1}{2} - 2^{p-1} \prod_{j=1}^{p} \epsilon_{i_j}$$

De aici putem calcula

$Pr[X_{i_1} \oplus \ldots \oplus X_{i_p} \oplus X_{i_{p+1}} = 0] =$

$$\left(\frac{1}{2} + 2^{p-1} \prod_{j=1}^{p} \epsilon_{i_j}\right)\left(\frac{1}{2} + \epsilon_{i_{p+1}}\right) + \left(\frac{1}{2} - 2^{p-1} \prod_{j=1}^{p} \epsilon_{i_j}\right)\left(\frac{1}{2} - \epsilon_{i_{p+1}}\right) = \frac{1}{2} + 2^p \prod_{j=1}^{p+1} \epsilon_{i_j}. \qquad \square$$

Corolarul 6.1. *Fie $\epsilon_{i_1,i_2,\ldots,i_k}$ tendinţa variabilei aleatoare $X_{i_1} \oplus \ldots X_{i_k}$.*

1. *Dacă $\exists j$ cu $\epsilon_{i_j} = 0$ atunci $\epsilon_{i_1,i_2,\ldots,i_k} = 0$.*
2. *Dacă $\epsilon_{i_j} = \pm 1/2$, $(j = 1, \ldots, k)$ atunci $\epsilon_{i_1,i_2,\ldots,i_k} = \pm 1/2$.*

Exemplul 6.4. *Să considerăm patru variabile aleatoare independente X_1, X_2, X_3, X_4. Fie $Pr[X_1 \oplus X_2 = 0] = 1/2 + \epsilon_{1,2}$ şi $Pr[X_2 \oplus X_3 = 0] = 1/2 + \epsilon_{2,3}$.*

Suma $X_1 \oplus X_3$ poate fi obţinută adunând $X_1 \oplus X_2$ cu $X_2 \oplus X_3$. Deci

$$Pr[X_1 \oplus X_3 = 0] = Pr[(X_1 \oplus X_2) \oplus (X_2 \oplus X_3) = 0].$$

Deci prin combinarea expresiilor liniare se vor obţine expresii liniare noi.

Considerând că variabilele aleatoare $X_1 \oplus X_2$ şi $X_2 \oplus X_3$ sunt independente, putem aplica lema Piling - Up, pentru a determina

$$Pr[X_1 \oplus X_3 = 0] = 1/2 + 2\epsilon_{1,2}\epsilon_{2,3}$$

Deci $\epsilon_{1,3} = 2\epsilon_{1,2}\epsilon_{2,3}$.

6.5.3 Relaţii de aproximare a S - boxurilor

Să folosim S - boxul 4×4 definit de Figura 1.

Vom lua ca exemplu expresia liniară $a_2 \oplus a_3 \oplus b_1 \oplus b_3 \oplus b_4 = 0$ sau – echivalent – $a_2 \oplus a_3 = b_1 \oplus b_3 \oplus b_4$, unde (a_1, a_2, a_3, a_4) este reprezentarea în binar a intrării din Figura 1, iar (b_1, b_2, b_3, b_4) este reprezentarea binară a ieşirii. Aplicând la intrare toate cele $2^4 = 16$ valori posibile şi examinând valorile de ieşire corespunzătoare, se poate observa (din Tabela 6) că expresia este verificată pentru 12 din cele 16 cazuri.

Deci tendinţa de probabilitate este $12/16 - 1/2 = 1/4$.

(a_1, a_2, a_3, a_4)	(b_1, b_2, b_3, b_4)	$a_2 \oplus a_3$	$b_1 \oplus b_3 \oplus b_4$	$a_1 \oplus a_4$	$a_3 \oplus a_4$	$b_1 \oplus b_4$
$(0,0,0,0)$	$(1,1,1,0)$	0	0	0	0	1
$(0,0,0,1)$	$(0,1,0,0)$	0	0	1	1	0
$(0,0,1,0)$	$(1,1,0,1)$	1	0	0	1	0
$(0,0,1,1)$	$(0,0,0,1)$	1	1	1	0	1
$(0,1,0,0)$	$(0,0,1,0)$	1	1	0	0	0
$(0,1,0,1)$	$(1,1,1,1)$	1	1	1	1	0
$(0,1,1,)$	$(1,0,1,1)$	0	1	0	1	0
$(0,1,1,1)$	$(1,0,0,0)$	0	1	1	0	1
$(1,0,0,0)$	$(0,0,1,1)$	0	0	1	0	1
$(1,0,0,1)$	$(1,0,1,0)$	0	0	0	1	1
$(1,0,1,0)$	$(0,1,1,0)$	1	1	1	1	0
$(1,0,1,1)$	$(1,1,0,0)$	1	1	0	0	1
$(1,1,0,0)$	$(0,1,0,1)$	1	1	1	0	1
$(1,1,0,1)$	$(1,0,0,1)$	1	0	0	1	0
$(1,1,1,0)$	$(0,0,0,0)$	0	0	1	1	0
$(1,1,1,1)$	$(0,1,1,1)$	0	0	0	0	1

Tabela 6: Exemple de aproximări liniare ale unui S - box

Similar, pentru relaţia $a_1 \oplus a_4 = b_2$ tendinţa de probabilitate este 0, iar pentru $a_3 \oplus a_4 = b_1 \oplus b_4$ tendinţa este $2/16 - 1/2 = -3/8$.

În general se pot construi $2^8 = 256$ expresii liniare cu (a_1, a_2, a_3, a_4) şi (b_1, b_2, b_3, b_4). Pentru fiecare din ele se poate da o reprezentare uniformă de tipul:

$$\left(\bigoplus_{i=1}^{4} x_i \cdot a_i\right) \oplus \left(\bigoplus_{i=1}^{4} y_i \cdot b_i\right)$$

unde $x_i, y_i \in \{0, 1\}$, $(1 \leq i \leq 4)$ iar operaţiile folosite sunt AND $(\cdot)$ şi XOR $(\oplus)$. Vectorii binari (x_1, x_2, x_3, x_4) – numiţi *sume de intrare* şi (y_1, y_2, y_3, y_4) (*sume de ieşire*) vor fi codificaţi prin cifre hexazecimale.

Astfel, fiecare din cele 256 expresii liniare se va scrie în mod unic ca o pereche de două cifre hexazecimale.

a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

Exemplul 6.5. *Expresia $a_1 \oplus a_4 \oplus b_2$ va avea suma de intrare $(1, 0, 0, 1)$ care este 9 în hexazecimal, iar suma de ieşire este $(0, 1, 0, 0)$, care este 4 în hexazecimal. Deci perechea ataşată variabilei este $(9, 4)$.*

Pentru o expresie având suma de intrare $x = (x_1, x_2, x_3, x_4)$ şi suma de ieşire $y = (y_1, y_2, y_3, y_4)$, fie $N_L(x, y)$ numărul octeţilor binari $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$ cu

$$\left(\bigoplus_{i=1}^{4} x_i \cdot a_i\right) \oplus \left(\bigoplus_{i=1}^{4} y_i \cdot b_i\right) = 0$$

Tendinţa unei astfel de expresii (alese aleator) este

$$\epsilon(x, y) = \frac{N_L(x, y) - 8}{16}.$$

O enumerare completă a tuturor tendinţelor pentru S - boxul folosit de sistemul nostru de criptare, este dată de *tabela de aproximări liniare* (pentru a evita fracţiile, nu s-a efectuat împărţirea la 16):

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2	0
6	0	+2	−2	+4	+2	0	0	+2	0	−2	+2	+4	−2	0	0	−2
7	0	−2	0	+2	+2	−4	+2	0	−2	0	+2	0	+4	+2	0	+2
8	0	0	0	0	0	0	0	0	−2	+2	+2	−2	+2	−2	−2	−6
9	0	0	−2	−2	0	0	−2	−2	−4	0	−2	+2	0	+4	+2	−2
A	0	+4	−2	+2	−4	0	+2	−2	+2	+2	0	0	+2	+2	0	0
B	0	+4	0	−4	+4	0	+4	0	0	0	0	0	0	0	0	0
C	0	−2	+4	−2	−2	0	+2	0	+2	0	+2	+4	0	+2	0	−2
D	0	+2	+2	0	−2	+4	0	+2	−4	−2	+2	0	+2	0	0	+2
E	0	+2	+2	0	−2	−4	0	+2	−2	0	0	−2	−4	+2	−2	0
F	0	−2	−4	−2	−2	0	+2	0	0	−2	+4	−2	−2	0	+2	0

Tabela 7: Tabela de aproximări liniare

Exemplul 6.6. *Tendinţa relaţiei liniare* $a_3 \oplus a_4 = b_1 \oplus b_4$ *(intrare hex* 3 *şi ieşire hex* 9*) este* $-6/16 = -3/8$*; deci probabilitatea ca această relaţie liniară să aibă loc este* $1/2 - 3/8 = 1/8$.

Din construcţia de mai sus se pot obţine imediat câteva proprietăţi:

- Probabilitatea ca pentru o intrare 0 să obţinem o anumită sumă nevidă de biţi de ieşire este 1/2 (deoarece pentru un S - box bijectiv, orice combinaţie liniară de biţi de ieşire trebuie să conţină un număr egal de 0 şi 1).
- Dacă suma biţilor de ieşire este 0, atunci suma biţilor de intrare este 0; deci o tendinţă +1/2 şi o valoare +8 în Tabela 7. În consecinţă, prima linie din Tabela 7 este 0, cu excepţia primului element. Aceeaşi proprietate o are şi prima coloană.
- Suma elementelor pe o linie sau coloană este ±8 (uşor de verificat).

6.5.4 Construirea aproximărilor liniare pentru un sistem bloc

Odată ce au fost stabilite aproximări liniare pentru S - boxurile unui SPN, putem merge mai departe, determinând o aproximare liniară de forma (1) pentru întreg sistemul de

criptare bloc. Aceasta se va realiza prin concatenarea aproximărilor liniare ale S -boxurilor consecutive.

După ce s-a obţinut o aproximare liniară bazată pe biţi din textul clar şi biţi care intră în ultima rundă a criptării, vom putea ataca sistemul găsind un sebset de biţi din cheia folosită în ultima rundă.

Ca şi la criptanaliza diferenţială, vom ilustra acest procedeu printr-un exemplu.

Să considerăm – conform Figurii 4 – aproximări liniare pentru S_{12}, S_{22}, S_{32} şi S_{34}. Anume:

S_{12} : $a_1 \oplus a_3 \oplus a_4 = b_2$ cu probabilitate 12/16 şi tendinţă +1/4;
S_{22} : $a_2 = b_2 \oplus b_4$ cu probabilitate 4/16 şi tendinţă −1/4;
S_{32} : $a_2 = b_2 \oplus b_4$ cu probabilitate 4/16 şi tendinţă −1/4;
S_{34} : $a_2 = b_2 \oplus b_4$ cu probabilitate 4/16 şi tendinţă −1/4.

Reamintim, $U_i(V_i)$ va reprezenta un bloc de 16 biţi care formează intrarea (ieşirea) din S - boxuri la runda i, iar $U_{i,j}(V_{i,j})$ va representa al j-lea bit din blocul $U_i(V_i)$ (biţii sunt numerotaţi de la 1 la 16).

Similar, K_i va reprezenta sub-cheia care se combină prin XOR cu biţii de intrare în runda i (o excepţie: sub-cheia K_5 se combină prin XOR cu biţii de ieşire din runda 4).

Deci $U_1 = P \oplus K_1$ unde $P = (P_1, \ldots, P_{16})$ reprezintă textul clar (de 16 biţi). Folosind aproximarea liniară de la prima rundă, avem

$$V_{1,6} = U_{1,5} \oplus U_{1,7} \oplus U_{1,8} = (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8}) \qquad (2)$$

cu probabilitate 3/4.

Aproximarea din runda 2 va da relaţia

$$V_{2,6} \oplus V_{2,8} = U_{2,6}$$

cu probabilitate 1/4. Deoarece $U_{2,6} = V_{1,6} \oplus K_{2,6}$, putem găsi o aproximare de forma

$$V_{2,6} \oplus V_{2,8} = V_{1,6} \oplus K_{2,6}$$

cu probabilitate 1/4.

Combinând această relaţie cu (2) (care are loc cu probabilitate 3/4), se obţine

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0 \qquad (3)$$

cu probabilitate $1/2 + 2(3/4 - 1/2)(1/4 - 1/2) = 3/8$ (deci o tendinţă −1/8) obţinută prin aplicarea Lemei 6.2.

Observaţia 6.4. *În toată această secţiune folosim prezumţia că aproximările S - boxurilor sunt independente; ipotetic, această ipoteză nu este foarte corectă, dar practic ea este valabilă pentru aproape toate sistemele de criptare bloc.*

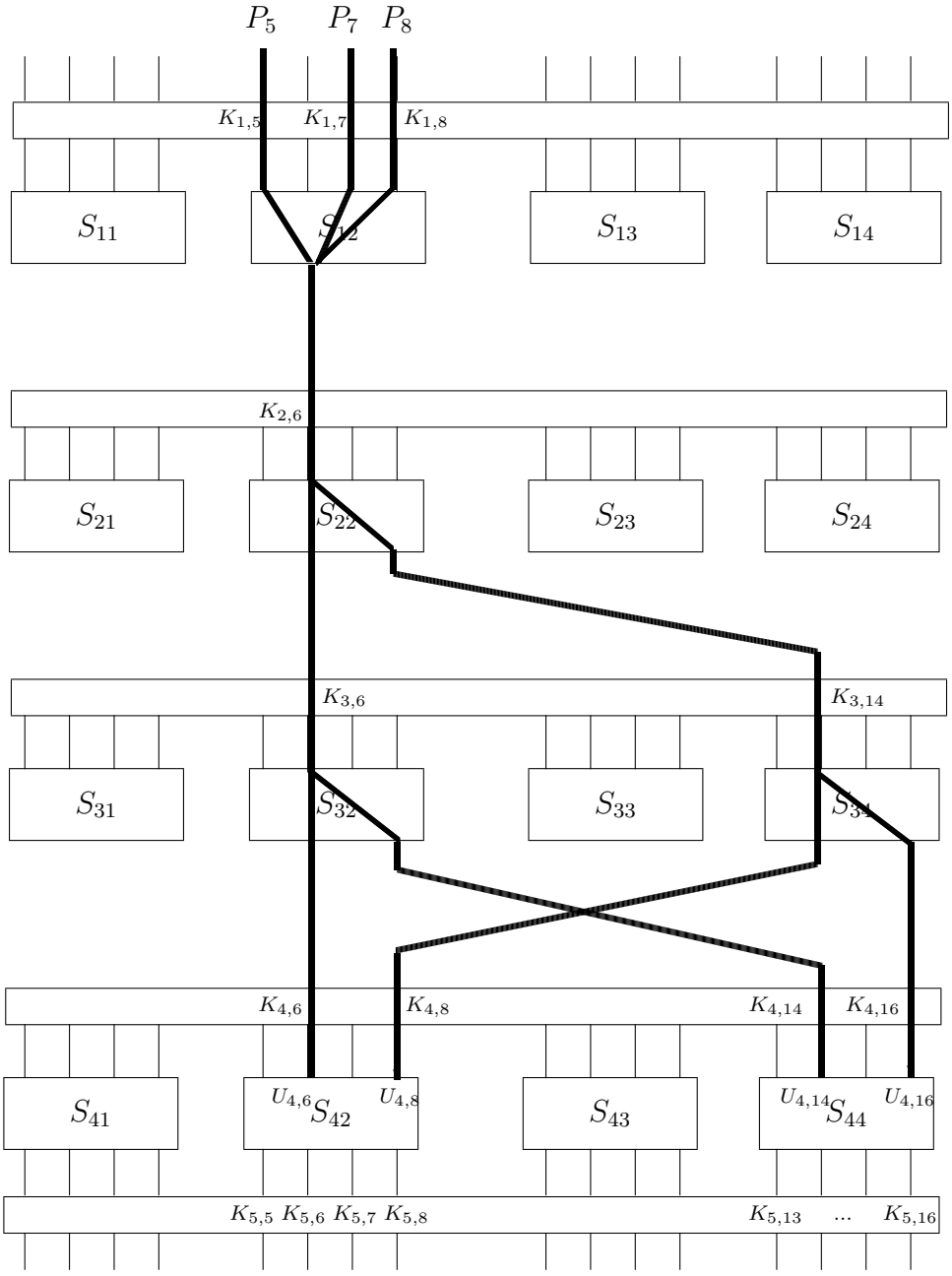


Figura 4: Exemplu de aproximare liniară

Pentru runda 3, reţinem că

$$V_{3,6} \oplus V_{3,8} = U_{3,6}$$

cu probabilitate 1/4 şi

$$V_{3,14} \oplus V_{3,16} = U_{3,14}$$

tot cu probabilitate 1/4.

Deci, pentru că $U_{3,6} = V_{2,6} \oplus K_{3,6}$ şi $U_{3,14} = V_{2,8} \oplus K_{3,14}$, aplicăm din nou Lema 6.2 şi obţinem

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus V_{2,6} \oplus K_{3,6} \oplus V_{2,8} \oplus K_{3,14} = 0 \quad (4)$$

cu probabilitate $1/2 + 2(1/4 - 1/2)2 = 5/8$ (deci, cu o tendinţă $+1/8$).

Combinăm acum relaţiile (3) şi (4) pentru a încorpora toate cele patru aproximări ale S - boxurilor. Vom avea

$$V_{3,6} \oplus V_{3,8} \oplus V_{3,14} \oplus V_{3,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} = 0.$$

Ţinem cont că $U_{4,6} = V_{3,6} \oplus K_{4,6}$, $U_{4,8} = V_{3,14} \oplus K_{4,8}$, $U_{4,14} = V_{3,8} \oplus K_{4,14}$, $U_{4,16} = V_{3,16} \oplus K_{4,16}$ şi putem rescrie această relaţie sub forma

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 \oplus \sigma_K = 0$$

unde

$\sigma_K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$

De remarcat că σ_K este o valoare fixată (0 sau 1), dependentă de cheia sistemului de criptare.

Aplicând Lema 6.2, această expresie are loc cu probabilitate $1/2+23(3/4-1/2)(1/4-1/2)3 = 15/32$ (deci o tendinţă $-1/32$).

Deoarece σ_K este fixat, deducem că relaţia

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \quad (5)$$

are loc cu probabilitate $15/32$ sau $(1 - 15/32) = 17/32$ (după cum $\sigma_K = 0$, respectiv 1).

Am obţinut deci o aproximare liniară a primelor trei runde ale sistemului de criptare, cu o tendinţă $1/32$, remarcabil de bună pentru construirea unui atac.

Să vedem cum o putem folosi această tendinţă pentru a determina o parte din biţii cheii.

6.5.5 Extragerea biţilor cheii

După ce am aflat o aproximare liniară pentru $N - 1$ runde dintr-un sistem de criptare cu N runde, aproximare având o tendinţă semnificativ de mare, putem ataca sistemul căutând biţi din ultima sub-cheie. În cazul nostru, având o aproximare liniară pentru 3 runde, este posibil să găsim biţi din subcheia K_5.

Vom numi *sub-cheie ţintă* setul de biţi din ultima sub-cheie, asociaţi S - boxurilor active din ultima rundă (S - boxuri influenţate de biţii utilizaţi în aproximarea liniară găsită). Pentru fiecare valoare posibilă a sub-cheii ţintă este definit un contor (iniţializat cu zero).

Procesul este similar celui folosit la criptanaliza diferenţială. Anume, pentru toate valorile posibile ale sub-cheii ţintă, biţii din textul criptat sunt XOR-aţi cu biţii din sub-cheia ţintă şi rezultatul este trecut în sens invers prin S - boxurile corespunzătoare.

Această procedură este efectuată pentru perechile *(text clar, text criptat)* folosite în atac. Dacă aproximarea liniară este verificată pentru biţii din textul clar folosit şi pentru rezultatul obţinut de procedură, contorul asociat sub-cheii respective este incrementat.

Sub-cheia ţintă al cărei contor diferă cel mai mult de jumătatea numărului de perechi de texte folosite, este cel mai probabil cea corectă. Aceasta rezultă din faptul că pentru sub-cheia ţintă respectivă, tendinţa este maximă. Dacă valoarea acestui contor este mai mare sau mai mică decât 1/2 depinde de faptul dacă aproximarea folosită este o relaţie liniară sau afină (lucru care depinde la rândul său de valorile necunoscute ale biţilor sub-cheii folosiţi implicit în aproximare).

Pentru o sub-cheie incorectă rezultată dintr-o alegere aleatoare, aproximarea asociată este verificată în general cu o probabilitate foarte apropiată de 1/2.

Exemplul 6.7. *([26]). Expresia liniară* (5) *afectează intrările în S - boxurile S_{42} şi S_{44} din ultima rundă. Deci, pentru fiecare pereche (text clar, text criptat) ales pentru atac, vom încerca toate cele $2^8 = 256$ valori ale sub-cheii ţintă $[K_{5,5}, \ldots, K_{5,8}, K_{5,13}, \ldots, K_{5,16}]$.*

Pentru fiecare valoare a sub-cheii ţintă vom incrementa contorul respectiv ori de câte ori relaţia (5) *este verificată (valorile $[U_{4,5}, \ldots, U_{4,8}, U_{4,13}, \ldots, U_{4,16}]$ sunt determinate trecând datele în sens invers prin sub-cheia ţintă şi S - boxurile S_{24} şi S_{44}).*

Contorul care diferă cel mai mult de jumătate din numărul de perechi (text clar, text criptat) folosite în atac va indica sub-cheia ţintă prezumtiv corectă. Valorile biţilor din sub-cheie folosiţi în calculul lui σ_K vor indica dacă tendinţa este pozitivă sau negativă. Când $\sigma_K = 0$, aproximarea liniară (5) *va servi ca estimare cu probabilitate $< 1/2$, iar când $\sigma_K = 1$, relaţia* (5) *va avea loc cu probabilitate $> 1/2$.*

$K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}$	$\lvert tendinta \rvert$	$K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}$	$\lvert tendinta \rvert$
1 C	0.0031	2 A	0.0044
1 D	0.0078	2 B	0.0186
1 E	0.0071	2 C	0.0094
1 F	0.0170	2 D	0.0053
2 0	0.0025	2 E	0.0062
2 1	0.0220	2 F	0.0133
2 2	0.0211	3 0	0.0027
2 3	0.0064	3 1	0.0050
2 4	**0.0336**	3 2	0.0075
2 5	0.0106	3 3	0.0162
2 6	0.0096	3 4	0.0218
2 7	0.0074	3 5	0.0052
2 8	0.0224	3 6	0.0056
2 9	0.0054	3 7	0.0048

Tabela 8. *Rezultate experimentale pentru un atac liniar*

În [26] este simulat un atac folosind 10000 *perechi cunoscute (text clar, text criptat) şi urmând procedura de criptare pentru valorile sub-cheii parţiale $[K_{5,5}, \ldots, K_{5,8}] =$* 0010 (*hex* 2) *respectiv $[K_{5,13}, \ldots, K_{5,16}] =$* 0100 (*hex* 4). *Contorul care diferă cel mai mult*

de valoarea 5000 *corespunde – aşa cum este de aşteptat – valorii sub-cheii ţintă* $[2, 4]$ *hex, confirmând că atacul a găsit corect biţii respectivi ai cheii.*

Tabela 8 *prezintă doar parţial rezultatele relative la contori (un tabel complet va avea* 256 *linii, câte una pentru fiecare valoare posibilă a sub-cheii ţintă). Valorile din tabel indică magnitudinea tendinţei, conform formulei* $|tendinta| = |cont - 5000|/10000$*, unde "cont" corespunde contorului asociat valorii respective date sub-cheii ţintă.*

După cum se vede, cea mai mare tendinţă apare pentru valoarea

$$[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}] = [2, 4],$$

rezultat care rămâne valabil pentru toată tabela cu 256 *valori.*

Valoarea 0.0336 *găsită experimental este foarte apropiată de valoarea estimată* $1/32 = 0.03125$*. De remarcat că în tabel apar şi alte valori destul de mari ale tendinţei (deşi teoretic, ele ar trebui să fie foarte apropiate de zero). Ele rezultă – printre altele – din alegerea aleatoare a perechilor de test, a proprietăţilor particulare ale* S *- boxurilor, a impreciziei ipotezei de independenţă (cerută de Lema 6.2).*

6.5.6 Criptanaliza liniară pentru DES

Cea mai bună aproximare liniară pentru un S - box din DES este

$$a_2 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4 = 0$$

asociată lui S_5. Această relaţie are probabilitatea 12/64.

Dacă se analizează componentele liniare ale funcţiei f, va rezulta că această relaţie conduce la aproximarea

$$a_{15} \oplus f(\alpha, K)_7 \oplus f(\alpha, K)_{18} \oplus f(\alpha, K)_{24} \oplus f(\alpha, K)_{29} \oplus K_{22} = 0 \qquad (6)$$

având tot probabilitatea 16/24. Relaţia (6) poate fi folosită cu succes pentru un atac contra sistemului DES cu 3 runde.

Pentru a ataca un DES complet (cu 16 runde) sunt folosite aproximările liniare:

$$\begin{cases} a_{15} \oplus f(X, K)_7 \oplus f(X, K)_{18} \oplus f(X, K)_{24} \oplus f(X, K)_{29} \oplus K_{22} = 0 \\ a_{27} \oplus a_{28} \oplus a_{30} \oplus a_{31} \oplus f(X, K)_{15} \oplus K_{42} \oplus K_{43} \oplus K_{45} \oplus K_{46} = 0 \\ a_{29} \oplus f(X, K)_{15} \oplus K_{44} = 0 \\ a_{15} \oplus f(X, K)_7 \oplus f(X, K)_{18} \oplus f(X, K)_{24} \oplus K_{22} = 0 \\ a_{12} \oplus a_{16} \oplus f(X, K)_7 \oplus f(X, K)_{18} \oplus f(X, K)_{24} \oplus K_{19} \oplus K_{23} = 0 \end{cases}$$

Atacul unui DES în 3 runde

Pentru atacul unui DES în 3 runde vom folosi structura din Figura 5 (care defineşte un astfel de sistem DES simplificat), împreună cu aproximarea (6).

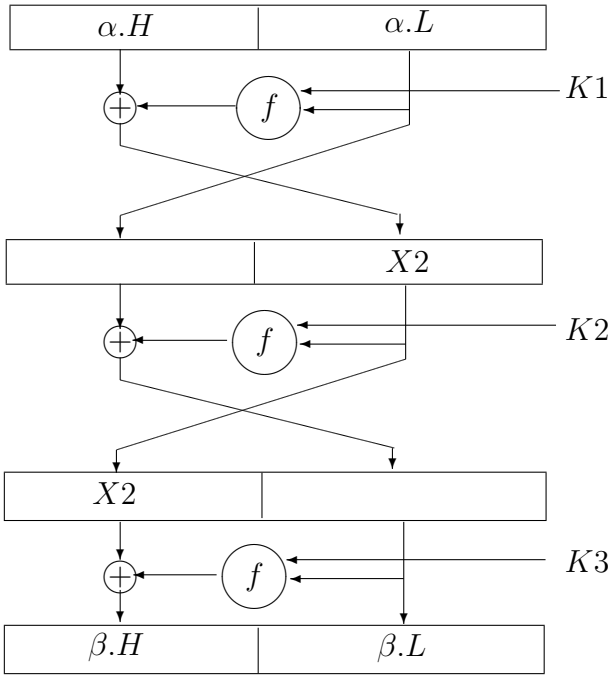


Figure 5: Structura unui DES cu 3 runde

Ele conduc la relaţia

$$\alpha.L_{15} \oplus (\alpha.H_7 \oplus X2_7) \oplus (\alpha.H_{18} \oplus X2_{18}) \oplus (\alpha.H_{24} \oplus X2_{24}) \oplus (\alpha.H_{29} \oplus X2_{29}) \oplus K1_{22} = 0$$

având probabilitatea tot 12/64 (s-a notat sub-cheia de la runda i cu Ki).

Bitul $K1_{22}$ din cheie este constant şi poate fi eliminat din relaţie (el doar face distincţia dintre valorile de probabilitate 12/64 sau $1 - 12/64$, diferenţă care nu ne interesează).

De remarcat că blocul $X2$ trece neschimbat prin runda a doua, deci va fi o intrare în ultima rundă. Rezultă că va trebui să parcurgem această rundă în sens invers şi să determinăm cei 4 biţi din $X2$ care apar în relaţia (6). Din Figura 5 avem

$$X2 = \beta.H \oplus f(\beta.L, K3),$$

unde singura necunoscută este sub-cheia $K3$.

Trebuie să găsim cei patru biţi $[X2_7, X2_{18}, X2_{24}, X2_{29}]$ folosind doar o sub-cheie. Dacă studiem funcţia f, vedem că aceşti biţi apar după o permutare şi o trecere prin S - boxuri. Cum sunt numai 4 biţi, ei nu pot ieşi decât din maxim 4 S - boxuri diferite. Pentru a calcula fiecare din aceste S - boxuri sunt necesari 6 biţi din cheie. Deci, în total sunt necesari 24 biţi din sub-cheia $K3$, pentru a găsi cei patru biţi din $X2$.

Folosind o căutare exhaustivă vom lua toate cele 2^{24} valori posibile ale biţilor din sub-cheie – pentru toate perechile *(text clar, text criptat)* – şi vom verifica dacă relaţia (6) este verificată. Valorile pentru care (6) se verifică în minim jumătate din cazuri, dau biţii probabil corecţi din $K3$.

Atacul unui DES în 16 runde

Pentru a ataca un sistem DES complet vor fi necesare toate cele cinci relaţii descrise anterior. Utilizarea lor ca aproximări ale funcţiilor f pentru un DES redus la 15 runde, conduce la aproximarea liniară:

$$\alpha.H[7,8,24] \oplus \alpha.L[12,16] \oplus \beta.H^{15}[7,18,24,29] \oplus \beta.L_{15}^{15} \oplus \sigma_{key} = 0 \quad (7)$$

unde $\beta.L^{15}$ şi $\beta.H^{15}$ sunt componentele stânga/dreapta ale textului criptat prin acest DES redus, iar σ_{key} este o valoare fixată, rezultată printr-un XOR între mai mulţi biţi de cheie din diverse runde. Conform cu Matsui, aceasta este cea mai bună aproximare a unui DES în 15 runde, iar tendinţa ei este 1.19^{-22}.

Putem extinde acest DES redus la un sistem DES normal, pe 16 runde, prin adăugarea unei ultime runde.

Deoarece runda 15 nu va mai fi ultima, va trebui să inversăm sub-blocurile $\beta.L^{15}$ şi $\beta.H^{15}$. Să notăm $\beta.H^{15} = X.L^{15}$ şi $\beta.L^{15} = X.H^{15}$.

Vom obţine din (7) următoarea aproximare (omiţând σ_{key}):

$$\alpha.H[7,8,24] \oplus \alpha.L[12,16] \oplus X.L^{15}[7,18,24,29] \oplus X.H_{15}^{15} = 0 \quad (8)$$

Relaţia (8) va fi aproximarea care trebuie verificată după trecerea în sens invers prin ultima rundă (a 16-a) din DES.

Fie $\beta.L$ şi $\beta.H$ componentele stânga/dreapta ale textului criptat obţinut după runda 16. Putem trece prin ultima rundă în sens invers, pentru a găsi blocurile $X.L^{15}$ respectiv $X.H^{15}$ de intrare în ultima rundă. Din construcţia sistemului DES avem:

$$X.L^{15} = \beta.L$$

şi

$$X.H^{15} = \beta.H \oplus f(X.L^{15}, K16) = \beta.H \oplus f(\beta.L, K16)$$

Înlocuind în (8), putem verifica dacă aproximarea respectivă are loc pentru o sub-cheie $K16$ dată, verificând pur şi simplu egalitatea

$$\alpha.H[7,18,24] \oplus \alpha.L[12,16] \oplus \beta.L[7,18,24,29] \oplus \beta.H_{15}^{15} \oplus f(\beta.L, K16)_{15} = 0$$

Singura componentă care nu este dată de o pereche *(text clar, text criptat)* folosită în atac, este bitul $f(\beta.L, K16)_{15}$. Acest bit este afectat de un singur S - box al lui f, deci sunt necesari numai 6 biţi din $K16$ pentru a verifica egalitatea (8) (remintim, S - boxurile DES folosesc intrări de 6 biţi).

Putem obţine în acest fel 6 biţi din cheie. Restul biţilor se vor obţine folosind eventual un atac prin forţă brută.

În [63] Matsui foloseşte simetria rundelor DES pentru a găsi altă aproximare liniară similară cu (7), având aceeaşi tendinţă. Pe baza ei se pot găsi alţi 6 biţi din cheie. Deci, în total pot fi descoperiţi 12 biţi din cheie. În [36], Matsui îmbunătăţeşte algoritmul,

folosind o aproximare pentru un DES pe 14 runde, obţinând în final 26 biţi din sub-cheia $K16$ (care are în total 48 biţi).

După estimarea lui Matsui, atacul originar necesită 2^{47} perechi *(text clar, text criptat)* pentru a avea succes cu probabilitate mare, iar atacul îmbunătăţit reduce acest număr la 2^{43} perechi. Cercetări experimentale ulterioare ([28]) sugerează că sunt suficiente 2^{41} perechi pentru ca atacul îmbunătăţit al lui Matsui să dea rezultate.

6.6 Comparare între criptanaliza diferenţială şi liniară

Între cele două tipuri de criptanaliză există multe similarităţi remarcabile. Astfel:

- Diferenţialele caracteristice corespund aproximărilor liniare. Tabelele de distribuţie a diferenţelor sunt înlocuite cu tabelele de aproximare liniară.
- Regula de combinare a diferenţialelor caracteristice: "substituie diferenţele şi înmulţeşte probabilităţile" corespunde regulii de combinare a aproximărilor liniare (Lema piling-up): "substituie elementele comune şi înmulţeşte tendinţele".
- Algoritmii de căutare pentru cea mai bună caracteristică sau cea mai bună aproximare liniară sunt în esenţă identici.

Cele mai importante distincţii între cele două metode de atac sunt:

- Criptanaliza diferenţială lucrează cu blocuri de biţi, în timp ce criptanaliza liniară lucrează în esenţă cu un singur bit.
- Tendinţa unei aproximări liniare este un număr cu semn. Deci, fiind date două aproximări cu aceleaşi structuri de intrare şi ieşire, aceiaşi probabilitate, dar semne opuse, aproximarea rezultată va avea tendinţa zero (datorită faptului că cele două aproximări se vor anula reciproc).

6.7 Exerciţii

6.1. *Fie S - boxul $\pi_S : \{0,1\}^3 \longrightarrow \{0,1\}^3$ definit prin tabela*

x	0	1	2	3	4	5	6	7
$\pi_S(x)$	0	2	3	4	5	6	7	1

Se dau două mesaje $s_1, s_2 \in \{0,1\}^3$. Să se afle cheia $k \in \{0,1\}^3$ ştiind că

$$s_1 \oplus k = 5, \qquad s_2 \oplus k = 3, \qquad \pi_S(s_1) \oplus \pi_S(s_2) = 2.$$

6.2. *Să se construiască aproximări liniare pentru S - boxul din exerciţiul precedent.*

6.3. *Fie X_1, X_2, X_3 variabile aleatoare independente cu valori în $\{0,1\}$ de tendinţe ϵ_1, ϵ_2 respectiv ϵ_3. Demonstraţi că $X_1 \oplus X_2$ şi $X_2 \oplus X_3$ sunt indepedente dacă şi numai dacă $\epsilon_1 = 0,\ \epsilon_3 = 0$ sau $\epsilon_2 = \pm 1/2$.*

6.4. *Pentru fiecare din cele opt S - boxuri DES calculaţi tendinţa variabilei aleatoare $X_2 \oplus Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$.*

6.5. *S - boxul DES S_4 are câteva proprietăţi specifice:*

1. *Arătaţi că a doua linie din S_4 poate fi obţinută din prima linie folosind operaţia*

$$(y_1, y_2, y_3, y_4) \longrightarrow (y_2, y_1, y_4, y_3) \oplus (0, 1, 1, 0)$$

2. *Arătaţi că orice linie din S_4 poate fi transformată în orice altă linie printr-o operaţie similară.*

6.6. *Fie $\pi_S : \{0,1\}^m \longrightarrow \{0,1\}^n$ un S - box. Demonstraţi că:*

1. $N_L(0,0) = 2^m$;
2. $N_L(a,0) = 2^m - 1, \quad \forall a \in [0, 2^m - 1]$;
3. $\sum_{a=0}^{2^m-1} \sum_{b=0}^{2^n-1} N_L(a,b) \in \{2^{n+2m-1}, 2^{n+2m-1} + 2^{n+m-1}\}$.

6.7. *Un S - box $\pi_S : \{0,1\}^m \longrightarrow \{0,1\}^n$ este "balansat" dacă*

$$|\pi_{s^{-1}}(y)| = 2^{n-m}$$

pentru orice $y \in \{0,1\}^n$.

Demonstraţi următoarele afirmaţii despre N_L pentru un S - box balansat:

1. $N_L(0,b) = 2^m - 1, \quad \forall b \in [0, 2^n - 1]$;
2. $\forall a \in [0, 2^m - 1], \quad \sum_{b=0}^{2^n-1} N_L(a,b) = 2^{m+n-1} - 2^{n-1} + i2^n$

   unde i este un număr întreg din intervalul $[0, 2^{m-n}]$.

6.8. *Fie S - boxul definită:*

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(x)$	8	4	2	1	C	6	3	D	A	5	E	7	F	B	9	0

Determinaţi tabela de aproximaţie liniară.

6.9. *Fie un sistem de criptare $\mathcal{P} = \mathcal{C} = \mathcal{K}$ care asigură confidenţialitate perfectă; deci, din $e_K(\alpha) = e_{K_1}(\alpha)$ rezultă $K = K_1$. Notăm $\mathcal{P}= Y = \{\beta_1, \beta_2, \ldots, \beta_N\}$. Fie α un bloc de text clar fixat. Definim funcţia $g : Y \longrightarrow Y$ prin $g(\beta) = e_\beta(\alpha)$. Definim un graf orientat Γ având a noduri Y şi ca arce $(\beta_i, g(\beta_i))$ $(1 \leq i \leq N)$.*

1. *Arătaţi că Γ este o reuniune de cicluri orientate disjuncte.*

2. *Fie T un parametru de timp fixat. Considerăm o mulţime $Z = \{\gamma_1, \ldots, \gamma_m\} \subseteq Y$, astfel ca pentru orice $\beta_i \in Y$, β_i este într-un ciclu de lungime cel mult T, sau există un element $\gamma_j \neq \beta_i$ astfel că distanţa de la β_i la γ_j (în Γ) este cel mult T. Demonstraţi că există o astfel de mulţime cu $card(Z) \leq 2N/T$ (deci $card(Z)$ este de complexitate $\mathcal{O}(N/T)$.*

3. *Pentru fiecare $\gamma - j \in Z$, definim $g^{-1}(\gamma_j)$ ca fiind acel element β_i astfel că $g^T(\beta - i) = \gamma_j$, unde g^T este funcţia g aplicată de T ori. Construiţi tabela X a perechilor $(\gamma_j, g^{-1}(\gamma_j))$, ordonate după prima coordonată.*

   Un algoritm care găseşte K astfel că $\beta = e_K(\alpha)$ este următorul:

```
1.  β_0 ← β;
2.  flag ← True;
3.  while g(β) ≠ β_0 do
    3.1.  if ∃ j β = γ_j and flag then
          3.1.1.  β ⟶ g^{-1}(γ_j)
          3.1.2.  flag ⟶ False
                  else
          3.1.3.  β ⟶ g(β);
    3.2.  K = β.
```

   Arătaţi că el determină K în maxim T etape (compromisul spaţiu - timp este deci $\mathcal{O}(N)$).

4. *Daţi un algoritm care construieşte o mulţime Z în timp $\mathcal{O}(NT)$, fără a folosi tablouri de mărine N.*

Bibliografie

[1] Anderson R. ş.a. - *Serpent: A proposal for the Advanced Encryption Standard*, http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/serpent.pdf

[2] Atanasiu A. - *Teoria codurilor corectoare de erori*, Editura Univ. Bucureşti, 2001;

[3] D. Bayer, S. Haber, W. Stornetta; Improving the efficiency and reliability of digital time-stamping. Sequences II, Methods in Communication, Security and Computer Science, Springer Verlag (1993), 329-334.

[4] E. Biham, A. Shamir, *Differential Cryptanalysis of DES - like Cryptosystems*, Journal of Cryptology, vol. 4, 1 (1991), pp. 3-72.

[5] E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[6] E. Biham, A. Shamir, *Differential Cryptanalysis of the Full 16-Round DES*, Proceedings of Crypto92, LNCS 740, Springer-Verlag.

[7] E. Biham, *On Matsuis Linear Cryptanalysis*, Advances in Cryptology - EUROCRYPT 94 (LNCS 950), Springer-Verlag, pp. 341-355, 1995.

[8] A. Biryukov, A. Shamir, D. Wagner, *Real Time Cryptanalysis of* $A5/1$ *on a PC*, Fast Software Encryption - FSE 2000, pp 118.

[9] A. Bruen, M. Forcinito, *Cryptography, Information Theory, and Error - Correction*, Wiley Interscience 2005.

[10] Bos J.N., Chaum D. - Provably unforgable signatures; Lecture Notes in Computer Science, $740(1993), 1 - 14$

[11] D. Chaum, H. van Antwerpen - Undeniable signatures; Lecture Notes in Computer Science, $435(1990), 212 - 216$

[12] D. Chaum, E. van Heijst, B. Pfitzmann; Cryptographically strong undeniable signatures, unconditionally secure for the signer. Lecture Notes in Computer Science, 576 (1992), 470-484.

[13] Brigitte Collard - *Secret Language in Graeco-Roman antiquity* (teză de doctorat) $http://bcs.fltr.ucl.ac.be/FE/07/CRYPT/Intro.html$

[14] Cook S., $http://www.claymath.org/millennium/P\_vs\_NP/Official\_Problem\_Description.pdf$

[15] Coppersmith D. ş.a. - *MARS - a candidate cypher for AES*, http://www.research.ibm.com/security/mars.pdf

[16] Daemen J., Rijmen V. - *The Rijndael Block Cipher Proposal*, http://csrc.nist.gov/CryptoToolkit/aes/

[17] I.B. Damgard; A design principle for hash functions. Lecture Notes in Computer Science, 435 (1990), 516-427.

[18] Diffie D.W., Hellman M.E. - *New Directions in Cryptography*, IEEE Transactions on Information Theory, IT-22, 6 (1976), pp. 644-654

[19] W. Diffie, M.E. Hellman - Multiuser cryptographic techniques; AFIPS Conference Proceedings, $45(1976), 109 - 112$

[20] L´ Ecuyer P. - *Random Numbers for Simulation*, Comm ACM 33, 10(1990), 742-749, 774.

[21] Enge A. - *Elliptic Curves and their applications to Cryptography*, Kluwer Academic Publ, 1999

[22] El Gamal T., *A public key cryptosystem and a signature scheme based on discrete algorithms*, IEEE Transactions on Information Theory, 31 (1985), 469-472

[23] Fog A. - *http://www.agner.org/random/theory*;

[24] Gibson J., *Discrete logarithm hash function that is collision free and one way.* IEEE Proceedings-E, 138 (1991), 407-410.

[25] S. Haber, W. Stornetta; How to timestamp a digital document. Journal of Cryptology, 3(1991), 99-111.

[26] H. M. Heyes, *A Tutorial on Linear and Differential Cryptanalysis.*

[27] van Heyst E., Petersen T.P. - How to make efficient fail-stop signatures; Lecture Notes in Computer Science, $658(1993), 366 - 377$

[28] P. Junod, *On the complexity of Matsuiş attack*, in SAC 01: Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography, pp 199211, London, UK, 2001. Springer-Verlag.

[29] Kahn D. - *The Codebreakers*, MacMillan Publishing Co, New York, 1967

[30] Kelly T. - *The myth of the skytale*, Cryptologia, Iulie 1998, pp. 244 - 260.

[31] A. Konheim - *Computer Security and Cryptography*, Wiley Interscience, 2007.

[32] Knuth D. - *The art of computer Programming*, vol 2 (Seminumerical Algorithms)

[33] Lenstra, H.W. - *Factoring Integers with Eiipltic Curves*, Annals of Mathematics, vol. 126, pp. 649-673, 1987.

[34] Matsui, M, Yamagishi, A. - *A new method for known plaintext attack of FEAL cipher.* Advances in Cryptology - EUROCRYPT 1992.

[35] M. Matsui - *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology - EUROCRYPT 93, LNCS 765, Springer-Verlag, pp. 386-397, 1994.

[36] M. Matsui - *The first experimental cryptanalysis of the Data Encryption Standard*, in Y.G. Desmedt, editor, Advances in Cryptology - Crypto 4, LNCS 839, SpringerVerlag (1994), 1- 11.

[37] M. Matsui - *New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptalaysis*, Fast Software Encryption, LNCS 1039, Springer-Verlag, 1996, pp. 205-218.

[38] Merkle, R., Hellman, M. - *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Trans. IT 24(5), Sept 1978, pp. 525530.

[39] Merkle R.C. - *A fast software one-way functions and DES*, Lecture Notes in Computer Science, 435 (1990), 428-446

[40] Mitchell C.J., Piper F., Wild, P. - Digital signatures; Contemporary Cryptology, The Science of Information Integrity, IEEE Press, (1992), $325 - 378$

[41] Menezes A., Oorschot P., Vanstome S., *Handbook of Applied Cryptography*

[42] B. Preneel, R. Govaerts, J. Vandewalle; Hash functions based on block ciphers: a syntetic approach. Lecture Notes in Computer Science, 773 (1994), 368-378

[43] Rivest R. ş.a - *The RC6^{TM} Block Cipher*,
ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pdf

[44] R.L. Rivest; The **MD4** message digest algorithm. Lecture Notes in Computer Science, 537, (1991), 303-311

[45] Rivest R., Shamir A., Adleman A., *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, Vol. 21 (2), 1978, pages 120–126.

[46] Rosing, M - *Implementing Elliptic Curve Cryptography*, Manning, 1998

[47] D. Salmon - *Data Privacy and Security*, Springer Professional Computing, 2003

[48] Salomaa A. - *Criptografie cu chei publice*, Ed. Militară, Bucureşti 1994

[49] Schneier B. - *Applied Cryptography*, John Wiley and Sons, 1995

[50] Schneier B ş.a. - *Twofish*, http://www.counterpane.com/twofish.html

[51] Shamir, A. - *A polynomial time Algorithm for breaking the basic Merkle - Hellman cryptosystem*,

http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C82/279.PDF

[52] Shoup, V. - *Lower bounds for discrete logarithm and related problems*, Advanced in Cryptology, EUROCRYPT 97, Springer - Verlag LNCS 1233, pp. 313-328, 1997.

[53] Selmer E.S. - *Linear Recurrence over Finite Field*, Univ. of Bergen, Norway, 1966;

[54] Sibley E.H. - *Random Number Generators: Good Ones are Hard to Find*, Comm ACM 31, 10(1988), 1192-1201.

[55] Smid M.E., Branstad, D.K. - *Response to comments on the $NIST$ proposed digital signature standard*, Lecture Notes in Computer Science, $740(1993), 76 - 88$

[56] Stinton D., *Cryptography, Theory and Practice*, Chapman& Hall/CRC, 2002

[57] Wiener M.J. - *Cryptanalysis of short RSA secret exponents*, IEEE Trans on Information Theory, 36 (1990), 553-558

[58] Williams H.C. - *Some public-key criptofunctions as intractable as factorisation*, Cryptologia, 9 (1985), 224-237.

[59] Zeng K.G., Yang C.H., Wei D.Y., Rao T.R.N.- *Pseudorandom Bit Generators in Stream Cipher Cryptography*, IEEE Computer, 24 (1991), 8.17.

[60] Secure hash Standard. National Bureau of Standards, FIPS Publications 180, 1993

[61] Digital signature standard; National Bureau of Standards, FIPS Publications 186, 1994

[62] $http://en.wikipedia.org/wiki/Enigma\_machine$

[63] *http* : *//en.wikipedia.org/wiki/M* − 209

[64] *http://en.wikipedia.org/wiki/Caesar_ cipher# History_ and_ usage*

[65] *http://psychcentral.com/psypsych/Polybius_ square*

[66] *http://www.answers.com/topic/vigen-re-cipher*

[67] *http://en.wikipedia.org/wiki/Rosetta_ stone*

[68] *Serpent homepage*, http://www.cl.cam.ac.uk/ rja14/serpent.html

[69] *P versus NP homepage*, http://www.win.tue.nl/ gwoegi/P-versus-NP.htm

[70] http://www.win.tue.nl/ gwoegi/P-versus-NP.htm

[71] http://en.wikipedia.org/wiki/Complexity_ classes_ P_ and_ NP