Simplified DES – en lättsmält version av DES

Simplied DES, eller S-DES, utvecklades 1996 av Edward F. Schaefer som en förenklad variant av DES anpassad för undervisning. I denna framställning skiljer sig några beteckingar från originalet.

Nödvändig algebra

Låt $\mathbb{B} = \{0, 1\}$ och låt $\mathbb{B}^n$ beteckna mängden av alla bitsträngar av längd n, där n är ett positivt heltal. Om $x, y \in \mathbb{B}$ så definierar vi $x \oplus y$ som det $z \in \mathbb{B}$ sådan att

$$x + y \equiv z \pmod{2}.$$

Denna operation ges av följande tabell.

$\oplus$	0	1
0	0	1
1	1	0

Sats 1. *Låt $x, y, z \in \mathbb{B}$. Då gäller följande.*

(a) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$

(b) $x \oplus y = y \oplus x$

(c) $x \oplus 0 = x$

(d) $x \oplus x = 0$

Låt $x, y \in \mathbb{B}^n$, d v s $x = (x_1 x_2 \ldots x_n)$ och $y = (y_1 y_2 \ldots y_n)$. Då definieras $x \oplus y$ som bitvis addition modulo 2 utan minne, d v s $x \oplus y = z$, där $z = (z_1 z_2 \ldots z_n)$ och $z_i = x_i \oplus y_i$ för varje i. Sats 1 kan generaliseras till $\mathbb{B}^n$, där 0 då betecknar den bitsträng av längd n som enbart består av 0:or.

Nyckelkonstruktion

Huvudnyckeln K består av 10 bitar, d v s

$$K = (k_0 k_1 \ldots k_9)$$

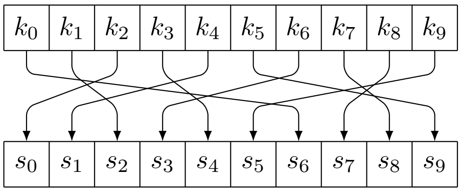
där varje k_i tillhör $\mathbb{B}$. Till kryptering och dekryptering behöver två rundnycklar

$$K_1 = (k_{1,0}, k_{1,1}, \ldots, k_{1,7}) \quad \text{och} \quad K_2 = (k_{2,0}, k_{2,1}, \ldots, k_{2,7}),$$

vilka består av 8 bitar vardera. Man utgår från huvudnyckeln K för att bestämma rundnycklarna K_1 och K_2. Låt

$$P_{10} = (2, 4, 1, 6, 3, 9, 0, 8, 7, 5)$$

beteckna en permutation av bitarna i k så att andra biten i k blir första biten efter permutationen, fjärde biten blir andra biten efter permutationen, o s v.



Med andra ord är

$$\begin{array}{lllll} s_0 = k_2 & s_1 = k_4 & s_2 = k_1 & s_3 = k_6 & s_4 = k_3 \\ s_5 = k_9 & s_6 = k_0 & s_7 = k_8 & s_8 = k_7 & s_9 = k_5 \end{array}$$

Därefter delar vi upp bitsträngen $(s_0s_1\dots s_9)$ i två delsträngar med 5 bitar vardera, d v s

$$(s_0s_1s_2s_3s_4) \quad \text{och} \quad (s_5s_6s_7s_8s_9).$$

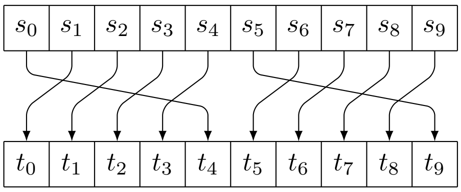
Varje delsträng roteras ett steg åt vänster, d v s

$$(s_1s_2s_3s_4s_0) \quad \text{och} \quad (s_6s_7s_8s_9s_5).$$

Konkatenering ger oss bitsträngen

$$(t_0t_1t_2t_3t_4t_5t_6t_7t_8t_8) = (s_1s_2s_3s_4s_0s_6s_7s_8s_9s_5).$$

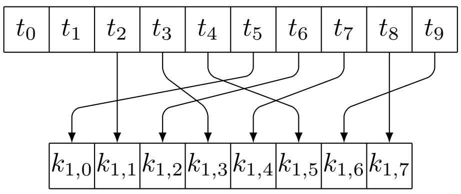
Ovanstående procedur med splittring, vänsterskift och konkatenering kan illustreras med följande figur.



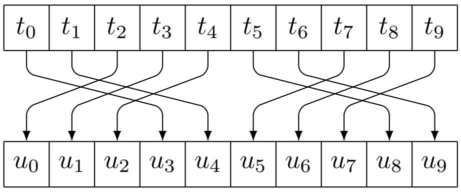
Låt

$$P_8 = (5, 2, 6, 3, 7, 4, 9, 8).$$

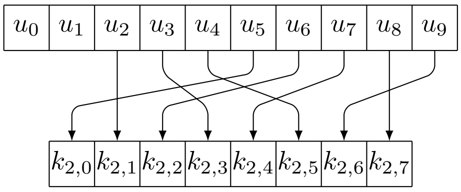
Första rundnyckeln K_1 erhålls genom att med P_8 i tur och ordning ta femte biten, andra biten, sjätte biten, o s v ur $(t_0t_1\dots t_9)$.



För att bestämma andra rundnyckeln K_2 fortsätter vi med $(t_0t_1\dots t_9)$. På liknande sätt som tidigare delare vi upp $(t_0t_1\dots t_9)$ i två delsträngar av vardera längd 5 och rotera varje delsträng två steg åt vänster innan konkatenering. Det ger oss en bitsträng $(u_0u_1\dots u_9)$. Processen kan illustreras enligt följande figur.

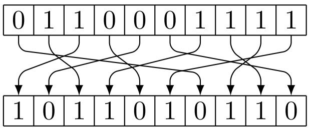


Avslutningsvis applicerar vi P_8 på $(u_0 u_1 \dots u_9)$.

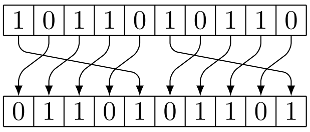


Det ger oss andra rundnyckeln K_2.

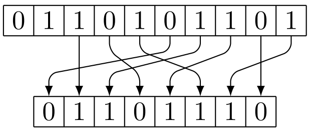
Exempel 1. Låt $K = 0110001111$. Permutationen P_{10} ger oss $s = 1011010110$, se följande figur.



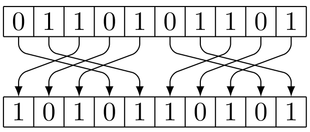
Rotation åt vänster ett steg av delsträngarna ger oss $t = 0110101101$, enligt nedanstående figur.



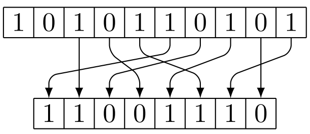
Funktionen P_8 ger första rundnyckeln $K_1 = 01101110$ enligt nedan.



Rotation åt vänster två steg av delsträngarna ger oss $u = 1010110101$, enligt nedanstående figur.



Funktionen P_8 ger oss nu också andra rundnyckeln $K_2 = 11001110$ enligt nedan.



Alltså är $K_1 = 01101110$ och $K_2 = 11001110$ de rundnycklar som fås med huvudnyckeln $K = 0110001111$.

Kryptering

Klartexten representeras som en bitsträng som delas in i block av längd 8. Låt

$$m = (m_0 m_1 m_2 m_3 m_4 m_5 m_6 m_7) \quad \text{och} \quad c = (c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7)$$

beteckna ett klartextblock respektive motsvarande kryptogramblock, där varje m_i och c_i tillhör $\mathbb{B}$. Om $E_K : \mathbb{B}^8 \to \mathbb{B}^8$ betecknar krypteringsfunktionen för S-DES, så är

$$c = E_K(m).$$

I detalj ges krypteirngfunktionen av

$$E_K = \mathrm{IP}^{-1} \circ F_2 \circ G \circ F_1 \circ \mathrm{IP},$$

d v s

$$c = \mathrm{IP}^{-1}(F_2(G(F_1(\mathrm{IP}(m))))).$$

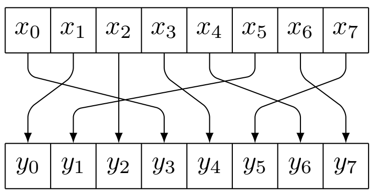
De ingående funktionerna IP, F_1, F_2 och G definieras nedan. Den *initiala permuationen* IP ges av

$$(1, 5, 2, 0, 3, 7, 4, 6).$$

På liknande sätt som i föregående avsnitt får vi att

$$(y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7) = \mathrm{IP}(x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7)$$

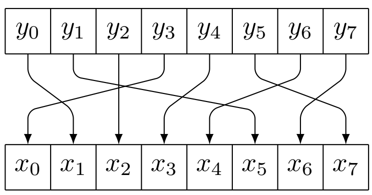
kan illustreras med nedanstående figur.



Funktionen IP är invertebar och dess invers IP^{-1} ges av

$$(3, 0, 2, 4, 6, 1, 7, 5).$$

Det ger oss följande figur.



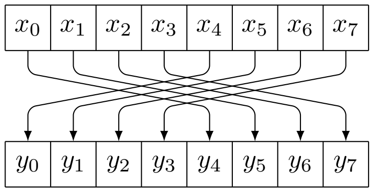
Låt $x = (x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7)$ vara ett block av längd 8. Vi inför skrivsättet

$$x = (L, R)$$

där $L = (x_0 x_1 x_2 x_3)$ och $R = (x_4 x_5 x_6 x_7)$, d v s L och R är vänster respektive höger delblock av x. Funktionen G byter plats på L och R. Med andra ord är

$$G(x) = G(L, R) = (R, L) = (x_4 x_5 x_6 x_7 x_0 x_1 x_2 x_3)$$

Alltså kan G beskrivas med följande figur.



Definitionen av funktionerna F_1 och F_2 är densamma med den skillnaden att rundnyckeln K_1 används i F_1 och rundnyckeln K_2 används i F_2. Funtionen F_i definieras enligt

$$F_i(x) = F_i(L, R) = (L \oplus f_i(R), R),$$

där funktionen $f_i \colon \mathbb{B}^4 \to \mathbb{B}^4$ är icke-inverterbar och som definieras enligt följande. Vi ska bestämma $f_i(R) = f_i(x_4x_5x_6x_7)$. Ställ upp nedanstående tabell.

$$\left[\begin{array}{c|cc|c} x_7 & x_4 & x_5 & x_6 \\ x_5 & x_6 & x_7 & x_4 \end{array}\right]$$

Addera sedan K_i till tabellen enligt föjande.

$$\left[\begin{array}{c|cc|c} x_7 \oplus k_{i,0} & x_4 \oplus k_{i,1} & x_5 \oplus k_{i,2} & x_6 \oplus k_{i,3} \\ x_5 \oplus k_{i,4} & x_6 \oplus k_{i,5} & x_7 \oplus k_{i,6} & x_4 \oplus k_{i,7} \end{array}\right]$$

Låt $a_{i,j}$ och $b_{i,j}$ beteckna resultatet av ovanstående enligt nedanstående tabell.

$$\left[\begin{array}{c|cc|c} a_{0,0} & b_{0,0} & b_{0,1} & a_{0,1} \\ a_{1,0} & b_{1,0} & b_{1,1} & a_{1,1} \end{array}\right]$$

Betrakta $(a_{i,0}a_{i,1})$ och $(b_{i,0}b_{i,1})$ som heltal i basen 2, d v s

$$a_i = 2a_{i,0} + a_{i,1} \quad \text{respektive} \quad b_i = 2b_{i,0} + b_{i,1}.$$

Låt p_i vara heltalet på rad a_i och kolumn b_i i S-box S_i, se tabell 1. Skriv om heltalen p_0 och p_1 i basen 2 med två bitar (lägg till en 0:a från vänster om p_i är 0 eller 1). Låt q_0, q_1, q_2, q_3 vara de bitar för vilka

$$p_0 = (q_0q_1) \quad \text{och} \quad p_1 = (q_2q_3).$$

Sätt $q = (q_0q_1q_2q_3)$. Slutligen, permutera bitarna i q med

$$P_4 = (1, 3, 2, 0),$$

d v s $(z_0z_1z_2z_3) = (q_1q_3q_2q_0)$. Då är

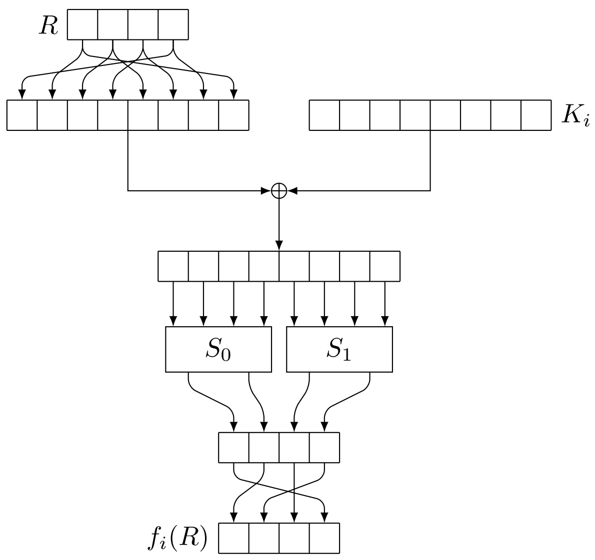
$$f_i(R) = f_i(x_4x_5x_6x_7) = (z_0z_1z_2z_3),$$

se figur 1.

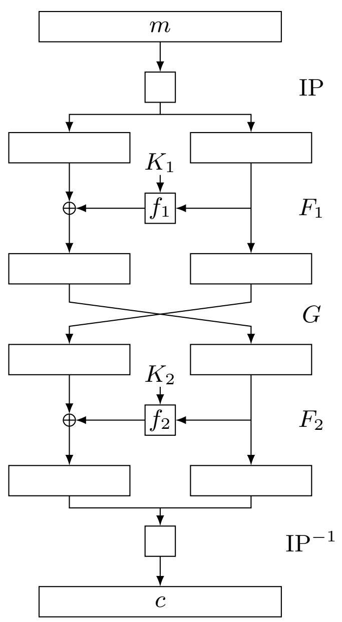
S_0	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S_1	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

Tabell 1. S-boxar för S-DES.



Figur 1. Funktionen f_i.



Figur 2. Kryperingsfunktionen i S-DES.

Alltså bestämmer man det kryptogram c som hör till klartexten m enligt följande algoritm.

[1] (Initial permutation IP) Sätt $y \leftarrow \mathrm{IP}(m)$ och låt L_0 och R_0 beteckna vänster respektive höger delblock i y, d v s $y = (L_0, R_0)$..

[2] (Funktionen F_1) Sätt $L_1 \leftarrow L_0 \oplus f_1(R_0)$ och $R_1 \leftarrow R_0$.

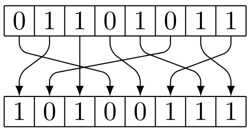
[3] (Funktionen G) Sätt $L_2 \leftarrow R_1$ och $R_2 \leftarrow L_1$.

[4] (Funktionen F_2) Sätt $L_3 \leftarrow L_2 \oplus f_2(R_2)$ och $R_3 \leftarrow R_2$.

[5] (Invers initial permutation IP^{-1}) Sätt $z \leftarrow (L_3, R_3)$ och $c \leftarrow \mathrm{IP}^{-1}(z)$.

Krypteringsfunktionen E kan i sin tur illustreras med figur 2.

Exempel 2. Låt $K = 0110001111$. Vi fann i exempel 1 att då ges rundnycklarna av $K_1 = 01101110$ och $K_2 = 11001110$. Antag att vi vill kryptera bokstaven `k` som binärt ges av $m = 01101011$, d v s heltalet 153. Första steget i krypteringen är permutationen IP. Vi får följande figur.



Alltså är $y = \mathrm{IP}(m) = 10100111$. Då är $L_0 = 1010$ och $R_0 = 0111$. Härnäst måste vi bestämma $f_1(R_0)$ för att kunna beräkna $F_1(y)$. Efterspm $R_0 = (x_4 x_5 x_6 x_7) = 0111$ så får vi uppställningen

$$\left[\begin{array}{c|cc|c} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{array}\right]$$

som ger i sin tur uppställningen

$$\left[\begin{array}{c|cc|c} 1 \oplus 0 & 0 \oplus 1 & 1 \oplus 1 & 1 \oplus 0 \\ 1 \oplus 1 & 1 \oplus 1 & 1 \oplus 1 & 0 \oplus 0 \end{array}\right] = \left[\begin{array}{c|cc|c} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

då vi adderar $K_1 = 01101110$. Det ger att

$$a_0 = 2 \cdot 1 + 1 = 3 \quad \text{och} \quad b_0 = 2 \cdot 1 + 0 = 2$$

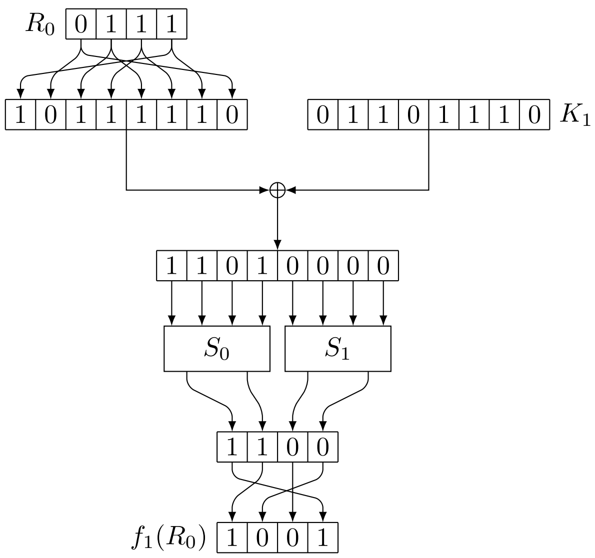
samt

$$a_1 = 2 \cdot 0 + 0 = 0 \quad \text{och} \quad b_1 = 2 \cdot 0 + 0 = 0.$$

Med andra ord ska vi avläsa rad 3 och kolumn 2 i S-box S_0 samt rad 0 och kolumn 0 i S-box S_1. Det ger oss $p_0 = 3$ och $p_1 = 0$, som binärt ges av 11 respektive 00. Alltså är $q = 1100$. Permutationen $P_4 = (1, 3, 2, 0)$ ger att

$$f_1(R_0) = f_1(0111) = 1001,$$

se nedanstående figur.



Det ger att

$$L_1 = L_0 \oplus f_1(R_0) = 1010 \oplus 1001 = 0011$$

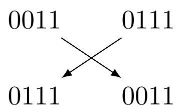
och

$$R_1 = R_0 = 0111.$$

Alltså är

$$F_1(10100111) = 00110111.$$

Nästa steg är att evaluera $G(00110111)$:



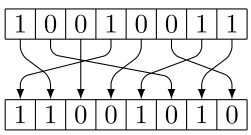
Alltså är $G(00110111) = 01110011$, vilket ger att $L_2 = 0111$ och $R_2 = 0011$. Nästa steg är att beräkna $F_2(01110011)$ och för det krävs att vi först bestämmer funktionsvärdet $f_2(R_2)$. Vi finner att $f_2(R_2) = 1110$, där detaljerna nämnas som övning. Det ger att

$$L_3 = L_2 \oplus f_2(R_2) = 0111 \oplus 1110 = 1001$$

och

$$R_3 = R_2 = 0011.$$

Det återstår att permutera (L_3, R_3) med IP^{-1} enligt följande figur.



Alltså är

$$c = E_K(m) = E_{0110001111}(01101011) = 11001010$$

det önskade kryptogrammet.

Dekryptering

Låt D_K beteckna dekrypteringsfunktionen för S-DES. Då är $D_K = E_K^{-1}$ under förutsättning att E_K är en inverterbar funktion. Vi definierade krypteringsfunktionen enligt

$$E_K = \mathrm{IP}^{-1} \circ F_2 \circ G \circ F_1 \circ \mathrm{IP},$$

d v s om $m \in \mathbb{B}^8$ är ett klartextblock så ges motsvarande kryptogramblock av

$$c = \mathrm{IP}^{-1}(F_2(G(F_1(\mathrm{IP}(m))))),$$

där $c \in \mathbb{B}^8$. Eftersom $\mathrm{IP}(\mathrm{IP}^{-1}(x)) = x$ för alla $x \in \mathbb{B}^8$, så är

$$\mathrm{IP}(c) = \mathrm{IP}(\mathrm{IP}^{-1}(F_2(G(F_1(\mathrm{IP}(m)))))) = F_2(G(F_1(\mathrm{IP}(m)))).$$

Låt x vara en godycklig bitsträng av längd 8 och sätt $y = F_i(x)$. Vi kan dela upp x och y i vänster och höger delblock, d v s $x = (L, R)$ och $y = (V, H)$ där $L, R, V, H \in \mathbb{B}^4$. Från definitionen av F_i följer att

$$V = L \oplus f_i(R) \quad \text{och} \quad H = R.$$

Det ger att

$$\begin{aligned} F_i(F_i(x)) &= F_i(y) \\ &= F_i(V, H) \\ &= (V \oplus f_i(H), H) \\ &= (V \oplus f_i(R), R) \\ &= ((L \oplus f_i(R)) \oplus f_i(R), R) \\ &= (L \oplus (f_i(R) \oplus f_i(R)), R) \\ &= (L \oplus 0000, R) \\ &= (L, R) \\ &= x, \end{aligned}$$

enligt sats 1. Vi har visat att F_i är sin egen invers, d v s $F_i^{-1} = F_i$. Alltså är

$$F_2(\mathrm{IP}(c)) = F_2(F_2(G(F_1(\mathrm{IP}(m))))) = G(F_1(\mathrm{IP}(m))).$$

Eftersom F byter plats på vänster och höger delblock har vi att

$$G(G(x)) = G(G(L, R)) = G(R, L) = (L, R) = x,$$

för alla $x \in \mathbb{B}^8$. Med andra ord är G också sin egen invers. Det ger att

$$G(F_2(\mathrm{IP}(c))) = G(G(F_1(\mathrm{IP}(m)))) = F_1(\mathrm{IP}(m))$$

och

$$F_1(G(F_2(\mathrm{IP}(c)))) = F_1(F_1(\mathrm{IP}(m))) = \mathrm{IP}(m)$$

Eftersom $\mathrm{IP}^{-1}(\mathrm{IP}(x)) = x$ för alla $x \in \mathbb{B}^8$, så har vi att

$$\mathrm{IP}^{-1}(F_1(G(F_2(\mathrm{IP}(c))))) = \mathrm{IP}^{-1}(\mathrm{IP}(m)) = m.$$

Vi har löst ut klartexten m och funnit att dekrypteringsfunktionen ges av

$$D_K = \mathrm{IP}^{-1} \circ F_1 \circ G \circ F_2 \circ \mathrm{IP}.$$

Notera ordningen på F_1 och F_2 jämfört med E_K.

Blockkryptering av längre klartexter

Antag att $E_K\colon \mathbb{B}^n \to \mathbb{B}^n$ är en krypteringsfunktion. Blocklängden är n enligt specifikationen av E_K. För att kunna kryptera en längre klartext delas denna in i block. Det finns flera olika metoder för hur man kan gå tillväga för att kryptera blocken (eng. *block cipher mode of operation*). Låt $m_1, m_2, \ldots$ och $c_1, c_2, \ldots$ beteckna klartext- respektive motsvarande kryptogramblock. Vidare definierar vi funktionerna $\mathcal{F}_m\colon \mathbb{B}^n \to \mathbb{B}^m$ och $\mathcal{L}_m\colon \mathbb{B}^n \to \mathbb{B}^m$, vilka returnerar de m första respektive de m sista bitarna i indata, där $1 \le m \le n$.

Electronic codebook (ECB)

Varje klartblock krypteras oberoende av andra enligt

$$c_i = E_K(m_i).$$

Dekryptering av respektive kryptogramblock ges i sin tur av

$$m_i = D_K(c_i).$$

Cipher Block Chaining (CBC)

Förutom nyckeln K väljer Alice och Bob ett fixt block c_0, som också betecknas IV, som är en förkortning för *initialvektor*. Innan kryptering av klartextblocket m_i adderas föregående kryptogramblock c_{i-1}, d v s

$$c_i = E_K(m_i \oplus c_{i-1}).$$

Dekryptering ges av

$$m_i = D_K(c_i) \oplus c_{i-1}.$$

Cipher Feedback (CFB)

Dela in klartexten i block av längd m, där $1 \le m \le n$. Alice och Bob väljer ett fixt block $x_1 \in \mathbb{B}^n$. Kryptering av blocken $m_1, m_2, \ldots$ ges då av

$$y_i = E_K(x_i), \quad c_i = m_i \oplus \mathcal{F}_m(y_i) \quad \text{och} \quad x_{i+1} = \mathcal{L}_n(x_i \parallel c_i),$$

där $x_i \| c_i$ betecknar konkatenering av bitsträngarna x_i och c_i. Notera att klartext- och kryptogramblock har längden m. Dekryptering ges av

$$y_i = E_K(x_i), \quad m_i = c_i \oplus \mathcal{F}_m(y_i) \quad \text{och} \quad x_{i+1} = \mathcal{L}_n(x_i \parallel c_i).$$

Output Feedback (OFB)

Dela in klartexten i block av längd m, där $1 \le m \le n$. Alice och Bob väljer ett fixt block IV $\in \mathbb{B}^n$. Kryptering ges då av

$$y_i = E_K(x_i), \quad c_i = m_i \oplus \mathcal{F}_m(y_i) \quad \text{och} \quad x_{i+1} = \mathcal{L}_n(x_i \parallel \mathcal{F}_m(y_i))$$

och dekryptering ges av

$$y_i = E_K(x_i), \quad m_i = c_i \oplus \mathcal{F}_m(y_i) \quad \text{och} \quad x_{i+1} = \mathcal{L}_n(x_i \parallel \mathcal{F}_m(y_i)).$$

Counter (CTR)

Dela in klartexten i block av längd m, där $1 \leq m \leq n$. Alice och Bob väljer ett fixt block $\mathrm{IV} \in \mathbb{B}^{n-j}$. Låt $\mathrm{bin}\colon \mathbb{Z}_{2^j} \to \mathbb{B}^j$ vara den funktion som bestämmer den binära representationen av ett element i $\mathbb{Z}_{2^j}$ så att bitsträngen består av j bitar med eventuella 0:or som utfyllnadstecken från vänster. Kryptering ges då av

$$x_i = \mathrm{IV} \parallel \mathrm{bin}(i), \quad y_i = E_K(x_i) \quad \text{och} \quad c_i = m_i \oplus \mathcal{F}_n(y_i)$$

och dekryptering ges av

$$x_i = \mathrm{IV} \parallel \mathrm{bin}(i), \quad y_i = E_K(x_i) \quad \text{och} \quad m_i = c_i \oplus \mathcal{F}_n(y_i).$$

Simplified DES i SageMath

Simplified DES är redan implementerad i SageMath. Det krävs att man laddar det programbibliotek som innehåller definitionen av Simplified DES.

```
sage: from sage.crypto.block_cipher.sdes import SimplifiedDES
sage: sDES = SimplifiedDES()
sage: sDES sdes
Simplified DES block cipher with 10-bit keys
```

Låt oss definiera den huvudnyckel K och klartext m som användes i exempel 2.

```
sage: K = sDES.string_to_list("0110001111")
sage: m = sDES.string_to_list("01101011")
sage: c = sDES.encrypt(m, K)
sage: sDES.list_to_string(c)
11001010
```

Vi får samma kryptogram som vi fann i exemplet. Dekryptering är lika enkelt.

```
sage: sDES.decrypt(c, K)
[0, 1, 1, 0, 1, 0, 1, 1]
```

Övriga funktioner knutna till Simplfied DES är följande.

`subkey(`K`)`	K_1
`subkey(`K`, 2)`	K_2
`left_shift(`B`, `n`)`	vänster rotation n steg
`initial_permutation(`B`)`	$\mathrm{IP}(B)$
`initial_permutation(`B`, inverse=True)`	$\mathrm{IP}^{-1}(B)$
`permutation10(`B`)`	$P_{10}(B)$
`permutation4(`B`)`	$P_8(B)$
`permutation8(`B`)`	$P_4(B)$
`permute_substitute(`B`, `K_i`)`	$F_i(B)$
`switch(`B`)`	$G(B)$
`sbox()`	$[S_1 S_2]$
`random_key()`	slumpmässigt vald nyckel

Referenser

[1] Edward F. Schaefer (1996), ”A Simplified Data Encryption Standard Algorithm, Cryptologia, 20:1, 77–84.

[2] The Sage Development Team, *Sage 9.2 Reference Manual: Cryptography*, 2020.