通信情報保護學會誌
第1卷, 第2號, 1991. 8

招請特輯

스트림(逐字) 暗號시스템에 關한 硏究
Study on the Stream Cipher Systems (2)

李 晩 榮*

創刊號에 이어 本稿에서는 同期逐字暗號시스템(synchronous stream ciphers)의 暗號化 및 復號, 키 自動키 暗號시스템, 그리고 自己同期逐字暗號시스템(self-synchronizing stream ciphers)의 暗號文歸還暗號시스템과 平文歸還暗號시스템에 關하여 分析하고자 한다.

目 次

* 종신회원, 漢陽大學校 名譽教授, 本 學會 會長

2.2 暗號化 및 復號

同期逐字暗號(synchronous stream cipher)는 키 數列(key-bit stream) $\mathbf{Z}=(z_0, z_1, z_2, z_3, \cdots)$가 平文과 獨立的으로 발생되는 시스템이다. m 段 LFSR의 탭 係數決定은 단지 2m 비트 만큼 키부호화키雙(key-encoded key pair)을 利用하면 쉽게 구할 수 있음을 前節에서 알았다. $\mathbf{Y}=(y_0, y_1, y_2, \cdots, y_{2m-1})$를 平文 $\mathbf{X}=(x_0, x_1, x_2, \cdots, x_{2m-1})$에 對應하는 暗號文이라고 하면 키 數列 $\mathbf{K}=(k_0, k_1, k_2, \cdots, k_{2m-1})$는 $x_i+y_i=x_i+(x_i+k_i)=k_i$, $0\leq i\leq 2m-1$, 를 計算함으로서 決定할 수 있다. 그러므로 萬若 2^m-1의 週期로 키 數列이 반복된다면 同期逐字暗號는 쉽게 깨질 수 있다. 키 數列이 쉽게 깨지지 않으려면, 平文만큼 긴 無作爲數列이어야 한다. 그러나 盜聽者는 2m 비트의 平文-暗號文雙으로 부터 쉽게 全體 키 數列을 유도해낼 수 있기 때문에 LFSR이 언제나 키 발생에 適合한 裝置라곤 볼 수 없다. 그러나 키 數列을 생성하는 m 段 LFSR은 수신단에서 暗號文을 平文으로 復號하기 爲한 키 數列이 再生될 수 있도록 唯一하게 결정되어야 한다. 다음의 例題를 通하여 同期逐字暗號의 暗號化와 復號化過程을 살펴보기로 하자.

[例題 4] 그림 5의 키 符號器를 利用하여 同期逐字暗號器를 構成해 보면 그림 8에 나타낸 것과 같다.

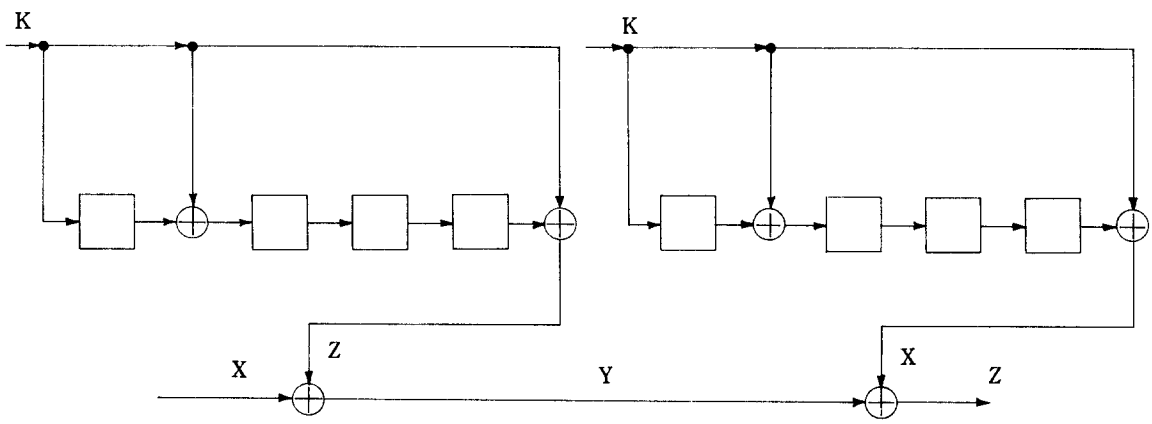

그림 8. 키 符號化를 利用한 逐字暗號 시스템

元來의 키 數列이 $\mathbf{K}=(11010001)$이라고 假定하면 對應되는 符號化된 키는 $\mathbf{Z}=(11000110)$이 된다. 平文을 $\mathbf{X}=(01001001)$로 假定하면 暗號文은 $\mathbf{Y}=\mathbf{X}+\mathbf{Z}=(10001111)$이 된다. 따라서 復號된 數列 $\mathbf{X}=\mathbf{Y}+\mathbf{Z}=(01001001)$는 再生된 平文이 된다.

[例題 5] 그림 8에 提示한 同期逐字暗號 시스템은 그림 9와 같이 다른 形態로 再構成할 수 있다. 그림 9를 가지고 그 宜當性을 理論的으로 分析해 보기로 하자.

스위치(switch)가 A-A′ 位置에 있을 때 그림은 入力이 平文 x(t)를 나타내고 出力이 暗號文 y(t)를 나타내는 暗號化 裝置가 된다. 반면에 스위치 位置가 B-B′일 때 그림은 入力이 暗號文 y(t)가 되고, 出力 u(t)는 再生된 平文 x(t)가 되는 復號化 裝置가 된다.

A-A′ 스위치雙(switch pair)을 이루는 LFSR은 暗號化裝置인데 그것을 表現하는 式은 다음과 같

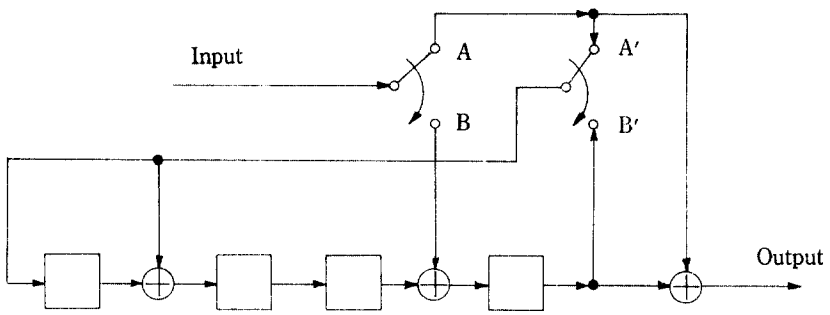

그림 9. LFSR을 利用한 暗號化 및 復號化 裝置

다.

$$y(t)=[(D+1)D^3+1]\,x(t)$$
$$=(D^4+D^3+1)\,x(t) \qquad (9)$$

여기서, LFSR은 遲延演算子 D로 表現되었다. 한편 스위치雙이 B−B일 때 LFSR로 構成된 復號化裝置는 다음과 같이 分析된다.

$$u(t)=[(D+1)D^2u(t)+y(t)]D$$
$$Dy(t)=(D^4+D^3+1)\,u(t) \qquad (10)$$

式(9)에 D를 곱하여 式(10)과 비교하면 다음과 같은 結果를 얻는다.

$$u(t)=Dx(t) \qquad (11)$$

따라서, 復號된 결과는 한 비트 遲延된 平文 x(t)와 同一하며 元來의 平文 x(t)는 한 비트 遲延된 u(t)로 正確하게 復元됨을 알 수 있다.

例題 4와 5에서 논의된 두個의 同期逐字暗號器를 比較해 보면, 그림 8의 暗號시스템은 두個의 같은 키 數列로 두 단계의 暗號化 및 復號化 過程으로 遂行되는 反面에 그림 9는 한個의 LFSR를 利用하여 暗號化 및 復號化 過程이 이루어짐을 보여주고 있다.

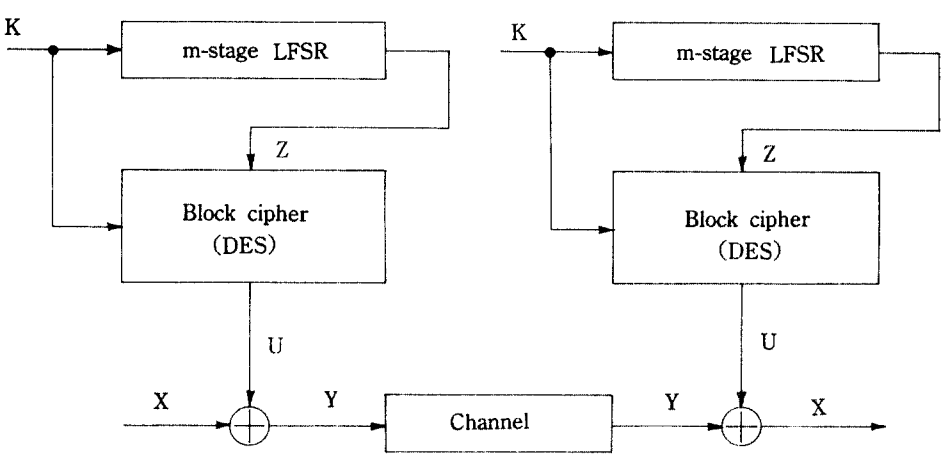

그림 10. LFSR과 DES를 結合시킨 同期 逐字暗號시스템

지금까지 논의한 바와 같이 단지 2m 비트의 平文-暗號文雙만 探知할 수 있다면 暗號 解讀者는 m 段 LFSR의 歸還係數(탭)를 容易하게 推定할 수 있을 뿐만 아니라 키 數列를 決定하게 되므로 線形 LFSR을 키 發生器로 使用하는 경우 週期性으로 因해 逐者暗號 시스템은 쉽게 깨질 수 있다. 따라서 이러한 弱點을 補完하기 爲하여 無週期性에 가까운 매우 긴 週期의 키 數列을 얻기 위해서는 LFSR의 出力側에 DES와 같은 블럭暗號裝置 또는 非線形 結合裝置(nonlinear combiner) 등을 導入함으로써 線形複雜度를 높이고 따라서 暗號解讀을 어렵게 만들 수 있을 것이다. 그러므로 키 數列의 內容을 推測不可能하게 만들기위해서는 그림 10과 같이 LFSR과 DES를 連鎖함으로써 키 數列을 擬似亂數化할 수 있을 것이다.

다음 例題는 그림 10을 利用하여 同期逐字暗號 시스템의 復號化過程을 풀어 보기로 하자.

[例題 6] m=32 段의 LFSR과 16 라운드 DES로 構成된 그림 10의 同期逐字暗號시스템을 생각해 보자. 만약 入力 키 **K**가 다음과 같다고 가정하면

$$\mathbf{K}=(5\,8\,1\,F\,B\,C\,9\,4\,D\,3\,A\,4\,5\,2\,E\,A) \quad \text{(hexadecimal)}$$
$$=(0101\ 1000\ 0001\ 1111\ 1011\ 1100\ 1001\ 0100\ 1101\ 0011\ 1010\ 0100\ 0101\ 0010\ 1110\ 1010) \quad \text{(binary)} \quad (12)$$

키 多項式은

$$K(x)=x+x^3+x^4+x^{11}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}+x^{18}+x^{19}+x^{20}+x^{21}+x^{24}+x^{27}+x^{29}+x^{32}+x^{33}+x^{35}+x^{38}+x^{39}+x^{40}+x^{42}+x^{45}+x^{49}+x^{51}+x^{54}+x^{56}+x^{57}+x^{58}+x^{60}+x^{62} \quad (13)$$

로 表現할 수 있다. 次數 32인 原始 多項式이 8進數나 2進數 表現으로 각각 다음과 같이 주어지므로

$$\mathbf{T}=(7\,0\,0\,0\,0\,0\,0\,2\,0\,0\,4) \quad \text{(octal)}$$
$$=(111\ 000\ 000\ 000\ 000\ 000\ 010\ 000\ 000\ 001) \quad \text{(binary)}$$

이에 對應하는 多項式은

$$T(x)=x^{32}+x^{22}+x^2+x+1 \quad (14)$$

로 된다. 이는 LFSR에서 탭 設定을 나타내는 特性方程式이다. LFSR로부터 出力 키 多項式 Z(x)는

$$Z(x)=K(x)T(x) \quad (15)$$

로 얻어진다. 式 (13)과 (14)를 利用하여 出力 키 多項式 Z(x)를 零이 아닌 項의 冪(power)으로 表現하면 아래와 같다.

$$\mathbf{Z}=(94\ 92\ 90\ 89\ 88\ 86\ 84\ 83\ 82\ 81\ 80\ 79\ 78\ 77\ 76\ 74\ 73\ 72\ 70\ 65\ 64\ 63\ 62\ 61\ 60\ 59\ 58\ 57\ 56\ 48\ 46\ 43\ 42\ 41\ 36\ 34\ 32\ 31\ 30\ 28\ 27\ 24\ 21\ 20\ 16\ 15\ 14\ 13\ 11\ 6\ 2\ 1)$$

x^i, $i>63$인 모든 項을 무시하면

$$\mathbf{Z}=(63\ 62\ 61\ 60\ 59\ 58\ 57\ 56\ 48\ 46\ 43\ 42\ 41\ 36\ 34\ 32\ 31\ 30\ 28\ 27\ 24\ 21\ 20\ 16\ 15\ 14\ 13\ 11\ 6\ 2\ 1) \quad (16)$$

또는

$$Z(x)=x+x^2+x^6+x^{11}+x^{13}+x^{14}+x^{15}+x^{16}+x^{20}+x^{21}+x^{24}+x^{27}+x^{28}+x^{30}+x^{31}+x^{32}+x^{34}+x^{36}+x^{41}+x^{42}+x^{43}+x^{46}+x^{48}+x^{56}+x^{57}+x^{58}+x^{59}+x^{60}+x^{61}+x^{62}+x^{63} \quad (17)$$

가 된다. Z(x)의 2進數 表現은

$$\mathbf{Z}=(0110\ 0010\ 0001\ 0111\ 1000\ 1100\ 1001\ 1011\ 1010\ 1000\ 0111\ 0010\ 1000\ 0000\ 1111\ 1111) \quad (18)$$

이고 16進數 表現은

$$\mathbf{Z}=(6\,2\,1\,7\,8\,C\,9\,B\,A\,8\,7\,2\,8\,0\,F\,F)$$

이다. DES의 키 스케쥴(key schedule)을 위해 式 (12)의 64비트 外部 키는 表 2에 따라 C_0(왼쪽 內容)와 D_0(오른쪽 內容) 두 부분으로 나누어지고 C_1과 D_1의 레지스터 內容은 블럭 C_0와 D_0를 왼쪽으로 한 비트 移動시킴으로써 얻어진다.

표 2. 交換 選擇表(PC-1)

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

표 3. 交換 選擇表(PC-2)

14	17	11	24	1	5	3	28	15	6	21	10		
23	19	12	4	26	8	16	7	27	20	13	2		
41	52	31	37	47	55	30	40	51	45	33	48		
44	49	39	56	34	53	46	42	50	36	29	32		

表 3 PC-2는 鎖狀化된(concatenated) 블럭 $(C_1, D_1), (C_2, D_2), \cdots, (C_{16}, D_{16})$으로부터 어떻게 內部 키 벡터 $K_1, K_2, \cdots, K_{16}$를 誘導해 내는가를 定義하는 法則이다. 실질적으로 DES의 48 비트 內部 키는 一連의 交換(series of permutation)과 64비트 內部 키로부터 만들어지는 56 키 비트의 왼쪽 置換(left shift)을 통해 發生된다. 그러므로 DES의 16개 內部 키 數列은 다음과 같이 계산된다.

K_1=(0010 0111 1010 0001 0110 1001 1110 0101 1000 1101 1101 1010)
K_2=(1101 1010 1001 0001 1101 1101 1101 0111 1011 0111 0100 1000)
K_3=(0001 1101 1100 0010 0100 1011 1111 1000 1001 0111 0110 1000)
K_4=(0010 0011 0101 1001 1010 1110 0101 1000 1111 1110 0010 1110)
K_5=(1011 1000 0010 1001 1100 0101 0111 1100 0111 1100 1011 1000)
K_6=(0001 0001 0110 1110 0011 1001 1010 1001 0111 1000 0111 1011)
K_7=(1100 0101 0011 0101 1011 0100 1010 0111 1111 1010 0011 0010)
K_8=(1101 0110 1000 1110 1100 0101 1011 0101 0000 1111 0111 0110)
K_9=(1110 1000 0000 1101 0011 0011 1100 0111 0101 0011 0001 0100)
K_{10}=(1110 0101 1010 1010 0010 1101 1101 0001 0010 0011 1110 1100)
K_{11}=(1000 0011 1011 0110 1001 0010 1111 0000 1011 1010 1000 1101)
K_{12}=(0111 1100 0001 1110 1111 0010 0111 0010 0011 1001 1010 1011)
K_{13}=(1111 0110 1111 0000 0100 1000 0011 1111 0011 1001 0111 0011)
K_{14}=(0000 1010 1100 0111 0101 0110 0010 0110 0111 1001 0111 0011)
K_{15}=(0110 1100 0101 1001 0001 1111 0110 0111 1010 1001 0111 0110)
K_{16}=(0100 1111 0101 0111 1010 0000 1100 0110 1100 0011 0101 1011)

표 4. E-비트 選定表

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

式 (18)에 表示된 LFSR로부터의 出力 키 **Z**는 逐字暗號시스템內 DES의 入力이 된다. DES의 入力 **Z**는 32 비트의 두 블럭 L_0(왼쪽)와 R_0(오른쪽)로 나누어진다. 32 비트 R_0는 表 4의 E-표(E bit-selection table)에 依해 48 비트로 擴大되고 뒤섞이게 된다. $E(R_0)$는 비트별로 K_1과 더해진다. 이 결과 얻어진 48 비트 $E(R_0)+K_1$은 S-box의 入力이 되고 非線形 S-box 變換을 통해서 32 비트 출력 B_1이 된다. 먼저 B_1의 비트들을 교환하고 $P(B_1)$과 L_0를 더하면 첫번째 라운드 후 右半쪽 出力 R_1이 된다.

$L_1=R_0$이므로 첫번째 라운드 후 左半쪽 出力은 즉시 다음과 같이 表現된다.
얻어진다. U_i를 各 라운드의 出力이라 할 때 U_i는

$$U_i=L_i*R_i,\ 1\leq i\leq 16 \qquad (19)$$

여기서 $L_i=R_{i-1}$이고 *는 鎖狀化를 나타내는 記號
이다. 그러므로 i=1인 첫번째 라운드의 出力은

U_1=(1101 1100 1011 0001 1001 1100 1010 1011 1010 0000 0111 0000 0101 1111 0011 0010)

이다. 마찬가지로 各各의 라운드 후 暗號化된 出力
U_2부터 U_{16}까지는 다음과 같이 쉽게 얻을 수 있
다.

U_2=(1010 0000 0111 0000 0101 1111 0011 0010 1001 0001 1111 0101 1101 1000 0010 0101)
U_3=(1001 0001 1111 0101 1101 1000 0010 0101 1010 0100 0001 1000 0001 0001 1111 1100)
U_4=(1010 0100 0001 1000 0001 0001 1111 1100 0110 1111 1010 1110 1110 0101 1111 1101)
U_5=(0110 1111 1010 1110 1100 0101 1111 1101 0110 0011 1111 0111 1101 0010 1011 0110)
U_6=(0110 0011 1111 0111 1101 0010 1011 0110 1110 0100 1011 0101 1101 1011 1111 0000)
U_7=(1110 0100 1011 0101 1101 1010 1110 1000 1100 0011 1101 0011 1101 1011 1111 0000)
U_8=(1100 0011 1101 0011 1101 1011 1111 0000 1001 1001 1111 0010 1100 0011 0110 0010)
U_9=(1001 1001 1111 0010 1100 0011 0110 0010 0001 1101 1111 0011 0100 1011 0100 1100)
U_{10}=(0001 1101 1111 0011 0100 1011 0100 1100 0000 0110 0000 1001 1001 0001 0011 0111)
U_{11}=(0000 0110 0000 1001 1001 0001 0011 0111 1000 1001 0111 1111 1111 0000 1001 1001)
U_{12}=(1000 1001 0111 1111 1111 0000 1001 1001 0000 1110 1001 1010 1011 1010 1011 0101)
U_{13}=(0000 1110 1001 1010 1011 1010 1011 0101 1011 1100 0001 0010 1101 1101 1001 0000)
U_{14}=(1011 1100 0001 0010 1101 1101 1001 0000 0110 0010 0000 1110 0000 0010 1011 1100)
U_{15}=(0110 0010 0000 1110 0000 0010 1011 1100 1100 1010 1000 0011 1011 0010 0101 1011)

그러나 先行 出力(preoutput) 64 비트 블럭은

$U_{16}=R_{16}*L_{16}$
=(1100 1010 1000 0011 1011 0010 0101 1011 0111 1111 0101 1001 1011 0100 1001 1101)

표 5. 逆 初期 交換表, IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

표 2. 交換 選擇表(PC-1)

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

표 3. 交換 選擇表(PC-2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

表 3 PC-2는 鎖狀化된(concatenated) 블럭 $(C_1, D_1), (C_2, D_2), \cdots, (C_{16}, D_{16})$으로부터 어떻게 內部 키 벡터 $K_1, K_2, \cdots, K_{16}$를 誘導해 내는가를 定義하는 法則이다. 실질적으로 DES의 48 비트 內部 키는 一連의 交換(series of permutation)과 64비트 內部 키로부터 만들어지는 56 키 비트의 왼쪽 置換(left shift)을 통해 發生된다. 그러므로 DES의 16개 內部 키 數列은 다음과 같이 계산된다.

K_1=(0010 0111 1010 0001 0110 1001 1110 0101 1000 1101 1101 1010)
K_2=(1101 1010 1001 0001 1101 1101 1101 0111 1011 0111 0100 1000)
K_3=(0001 1101 1100 0010 0100 1011 1111 1000 1001 0111 0110 1000)
K_4=(0010 0011 0101 1001 1010 1110 0101 1000 1111 1110 0010 1110)
K_5=(1011 1000 0010 1001 1100 0101 0111 1100 0111 1100 1011 1000)
K_6=(0001 0001 0110 1110 0011 1001 1010 1001 0111 1000 0111 1011)
K_7=(1100 0101 0011 0101 1011 0100 1010 0111 1111 1010 0011 0010)
K_8=(1101 0110 1000 1110 1100 0101 1011 0101 0000 1111 0111 0110)
K_9=(1110 1000 0000 1101 0011 0011 1100 0111 0101 0011 0001 0100)
K_{10}=(1110 0101 1010 1010 0010 1101 1101 0001 0010 0011 1110 1100)
K_{11}=(1000 0011 1011 0110 1001 0010 1111 0000 1011 1010 1000 1101)
K_{12}=(0111 1100 0001 1110 1111 0010 0111 0010 0011 1001 1010 1011)
K_{13}=(1111 0110 1111 0000 0100 1000 0011 1111 0011 1001 0111 0011)
K_{14}=(0000 1010 1100 0111 0101 0110 0010 0110 0111 1001 0111 0011)
K_{15}=(0110 1100 0101 1001 0001 1111 0110 0111 1010 1001 0111 0110)
K_{16}=(0100 1111 0101 0111 1010 0000 1100 0110 1100 0011 0101 1011)

표 4. E-비트 選定表

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

式 (18)에 表示된 LFSR로부터의 出力 키 **Z**는 逐字暗號시스템內 DES의 入力이 된다. DES의 入力 **Z**는 32 비트의 두 블럭 L_0(왼쪽)와 R_0(오른쪽)로 나누어진다. 32 비트 R_0는 表 4의 E-표(E bit-selection table)에 依해 48 비트로 擴大되고 뒤섞이게 된다. $E(R_0)$는 비트별로 K_1과 더해진다. 이 결과 얻어진 48 비트 $E(R_0)+K_1$은 S-box의 入力이 되고 非線形 S-box 變換을 통해서 32 비트 출력 B_1이 된다. 먼저 B_1의 비트들을 교환하고 $P(B_1)$과 L_0를 더하면 첫번째 라운드 후 右半쪽 出力 R_1이 된다.

$L_1=R_0$이므로 첫번째 라운드 후 左半쪽 出力은 즉시 얻어진다. U_i를 各 라운드의 出力이라 할 때 U_i는 다음과 같이 表現된다.

$$U_i=L_i * R_i,\ 1\leq i\leq 16 \qquad (19)$$

여기서 $L_i=R_{i-1}$이고 *는 鎖狀化를 나타내는 記號이다. 그러므로 i=1인 첫번째 라운드의 出力은

U_1=(1101 1100 1011 0001 1001 1100 1010 1011 1010 0000 0111 0000 0101 1111 0011 0010)

이다. 마찬가지로 各各의 라운드 후 暗號化된 出力 U_2부터 U_{16}까지는 다음과 같이 쉽게 얻을 수 있다.

U_2=(1010 0000 0111 0000 0101 1111 0011 0010 1001 0001 1111 0101 1101 1000 0010 0101)
U_3=(1001 0001 1111 0101 1101 1000 0010 0101 1010 0100 0001 1000 0001 0001 1111 1100)
U_4=(1010 0100 0001 1000 0001 0001 1111 1100 0110 1111 1010 1110 1110 0101 1111 1101)
U_5=(0110 1111 1010 1110 1100 0101 1111 1101 0110 0011 1111 0111 1101 0010 1011 0110)
U_6=(0110 0011 1111 0111 1101 0010 1011 0110 1110 0100 1011 0101 1101 1011 1111 0000)
U_7=(1110 0100 1011 0101 1101 1010 1110 1000 1100 0011 1101 0011 1101 1011 1111 0000)
U_8=(1100 0011 1101 0011 1101 1011 1111 0000 1001 1001 1111 0010 1100 0011 0110 0010)
U_9=(1001 1001 1111 0010 1100 0011 0110 0010 0001 1101 1111 0011 0100 1011 0100 1100)
U_{10}=(0001 1101 1111 0011 0100 1011 0100 1100 0000 0110 0000 1001 1001 0001 0011 0111)
U_{11}=(0000 0110 0000 1001 1001 0001 0011 0111 1000 1001 0111 1111 1111 0000 1001 1001)
U_{12}=(1000 1001 0111 1111 1111 0000 1001 1001 0000 1110 1001 1010 1011 1010 1011 0101)
U_{13}=(0000 1110 1001 1010 1011 1010 1011 0101 1011 1100 0001 0010 1101 1101 1001 0000)
U_{14}=(1011 1100 0001 0010 1101 1101 1001 0000 0110 0010 0000 1110 0000 0010 1011 1100)
U_{15}=(0110 0010 0000 1110 0000 0010 1011 1100 1100 1010 1000 0011 1011 0010 0101 1011)

그러나 先行 出力(preoutput) 64 비트 블럭은

$U_{16}=R_{16} * L_{16}$
=(1100 1010 1000 0011 1011 0010 0101 1011 0111 1111 0101 1001 1011 0100 1001 1101)

표 5. 逆 初期 交換表, IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

로 計算된다. 이 先行 出力된 데이타 블럭은 表 5 IP^{-1}에 依據해서 遂行함으로서 最終 키 비트 數列 U를 얻게 된다.

$$U=(0111\ 0011\ 1110\ 1010\ 0100\ 0101\ 1101\ 0011\ 0101\ 1111\ 0100\ 1100\ 1101\ 0010\ 1010\ 1101)$$
$$=(7\ 3\ E\ A\ 4\ 5\ D\ 3\ 5\ F\ 4\ C\ D\ 2\ A\ D)$$
(hexadeximal) (20)

64 비트 平文

$$X=(9\ A\ 2\ 6\ F\ 4\ 6\ 5\ D\ C\ 2\ 6\ B\ E\ 1\ 8)$$
$$=(1001\ 1010\ 0010\ 0110\ 1111\ 0100\ 0101\ 0011\ 1101\ 1100\ 0010\ 0110\ 1011\ 1110\ 0001\ 1000)$$

를 式(20)에서 얻은 64 비트 키 數列 U로 제어하여 64 비트의 暗號文 Y로 暗號化시키는 경우를 생각해 보자. 이 때 暗號文 Y는 X와 U의 비트별 덧셈에 의해 다음과 같이 求할 수 있다.

$$Y=X+U$$
$$=(1110\ 1001\ 1100\ 1100\ 1011\ 0001\ 1011\ 0110\ 1000\ 0011\ 0110\ 1110\ 0110\ 1100\ 1011\ 0101)$$
$$=(E\ 9\ C\ C\ B\ 1\ B\ 6\ 8\ 3\ 6\ E\ 6\ C\ B\ 5) \quad (21)$$

2元 加算을 통해 復號는 매우 간단히 이루어진다. 暗號文 Y에 같은 키 비트 數列 U를 더하면 X=U+Y가 되어 平文 X는 復元될 수 있는 것이다.

2.3 키 自動키 同期 暗號 시스템

同期 逐字 暗號의 특수한 한 形態는 키 自動 키 시스템으로서 그림 11에서 보듯이 키 비트 數列로부터 歸還이 얻어지는 것이다. 키 自動 키 逐字 暗號는 復號 過程에서 오류(error) 傳播가 없다. 왜냐하면 暗號文 Y에 있는 오류는 오직 복원된 平文 X에서 對應되는 비트 位置에만 오류를 發生시키기 때문이다.

그림 11에서 $S=(s_1, s_2, \cdots, s_m)$는 레지스터의 初期값이고 $T=(g_1, g_2, \cdots, g_m)$는 탭 係數이다. 平文을 $X=(x_0, x_1, \cdots, x_i, \cdots)$, 키 비트 數列을 $K=(k_0, k_1, \cdots, k_i, \cdots)$, 그리고 暗號文을 $Y=(y_0, y_1, \cdots, y_i, \cdots)$라 하면, 키 K를 使用하므로서 暗號化 및 復號는

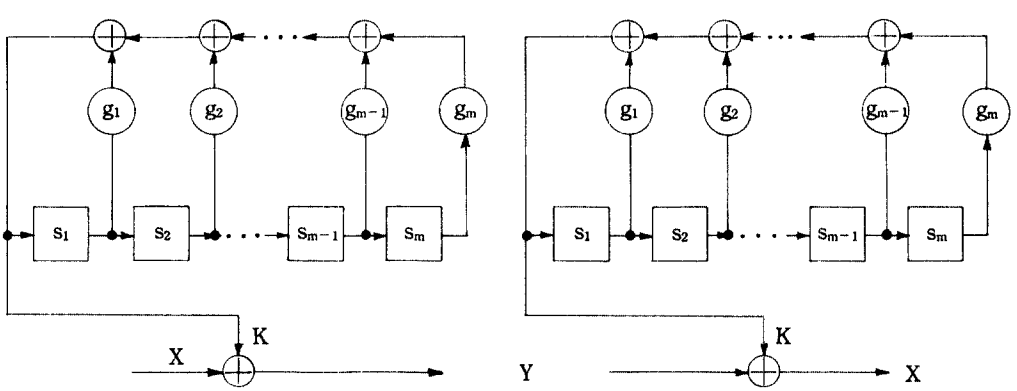

그림 11. m段 LFSR로 構成된 키 自動키 暗號시스템

다음과 같이 정의된다.

$$\begin{aligned} y_i &= x_i + k_i \\ x_i &= y_i + k_i \end{aligned} \quad , \ i=0,1,2,\cdots \tag{22}$$

여기서 x_i와 y_i는 各各 i번째의 平文과 暗號文 비트를 나타낸다. 그림 11을 참조하면 키 數列 **K**로부터 키 비트 k_i, $0 \leq i \leq m$,는 다음과 같이 表現된다.

$$\begin{aligned} k_0 &= g_m s_m + g_{m-1} s_{m-1} + \cdots + g_2 s_2 + g_1 s_1 \\ k_1 &= g_m s_{m-1} + g_{m-1} s_{m-2} + \cdots + g_2 s_1 + g_1 k_0 \\ &\vdots \qquad\qquad\qquad \vdots \\ k_{m-1} &= g_m s_1 + g_{m-1} k_0 + \cdots + g_2 k_{m-3} + g_1 k_{m-2} \\ k_m &= g_m k_0 + g_{m-1} k_1 + \cdots + g_2 k_{m-2} + g_1 k_{m-1} \end{aligned} \tag{23}$$

式 (23)의 키 비트에 대한 行列 表現은

$$\begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_{m-1} \\ k_m \end{bmatrix} = \begin{bmatrix} s_m & s_{m-1} & \cdots & s_2 & s_1 & g_m \\ s_{m-1} & s_{m-2} & \cdots & s_1 & k_0 & g_{m-1} \\ & & \vdots & & & \vdots \\ s_1 & k_0 & \cdots & k_{m-3} & k_{m-2} & g_2 \\ k_0 & k_1 & \cdots & k_{m-2} & k_{m-1} & g_1 \end{bmatrix} \tag{24}$$

이고 式 (23)은 또한 아래와 같이 簡略하게 表現할 수도 있다.

$$k_i = \begin{cases} \sum_{t=1}^{i} g_t k_{i-t} + \sum_{t=i+1}^{m} g_t s_{t-i} \ , & 0 \leq i \leq m-1 \\ \sum_{t=1}^{m} g_t k_{i-t}, & m \leq i \end{cases} \tag{25}$$

일단 키 비트 k_i를 決定하면 式(25)를 式(22)에 代入하여 暗號化와 復號化 變換을 쉽게 얻을 수 있다.

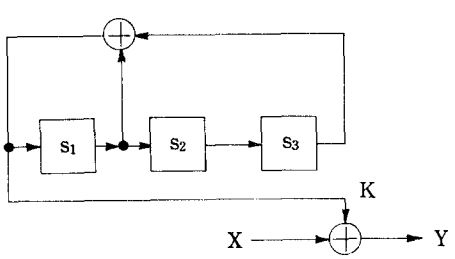

그림 12. 3段 LFSR을 利用한 키 自動키 暗號器

[例題 7] (7,4) Hamming 符號의 生成多項式 $g(x)=1+x+x^3$을 그림 12에 보인 LFSR의 탭 多項式 T(x)로 使用하고 초기 씨드 벡터(initial seed vector)가 **S**=(100)일 때 키 비트 **K**=(k_0, k_1, k_2, k_3)를 決定해 보자. 이 문제에 式 (24)를 適用하면

$$\begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} s_3 & s_2 & s_1 \\ s_2 & s_1 & k_0 \\ s_1 & k_0 & k_1 \\ k_0 & k_1 & k_2 \end{bmatrix} \begin{bmatrix} g_3 \\ g_2 \\ g_1 \end{bmatrix} \tag{26}$$

가 얻어지고 初期 씨드 벡터와 탭 係數를 式 (26)에 대입하면 다음과 같다.

$$\begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & k_0 \\ 1 & k_0 & k_1 \\ k_0 & k_1 & k_2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

이것으로부터

$$\begin{aligned} k_0 &= 1 \\ k_1 &= k_0 \\ k_2 &= 1 + k_1 \\ k_3 &= k_0 + k_2 \end{aligned}$$

를 얻을 수 있다. 따라서 키 비트 $k_0=1$, $k_1=1$, $k_2=0$, $k_3=1$, 즉 **K**=(1101)을 決定할 수 있다. 키 비트 **K**를 구하였으므로, 平文 數列 **X**가 주어지면 式 (22)를 使用하여 暗號化와 復號化를 遂行하는 演算을 쉽게 處理할 수 있다.

키 비트 數列 **K**를 生成하는 또 다른 方法을 찾아보자. 그림 11에 구현된 LFSR을 보면 탭 生成多項式(때로는 特性 多項式이라 한다.) $T(x)=1+g_1x+g_2x^2+\cdots+g_mx^m$으로 表現된다. 여기서 $g_i \in GF(2)$는 탭 係數(또는 歸還係數)이다. 만일 $s_i(t)$가 t번째 펄스 후의 LFSR의 i번째 값을 나타낸다면, $s_i(t+1)=s_{i-1}(t)$, $i=1,2,\cdots,m$로 表現될 수 있다.

따라서 그림 11의 LFSR을 보면, 다음과 같은 關係式을 求할 수 있다.

$$
\begin{aligned}
s_m(t+1) &= s_{m-1}(t) \\
s_{m-1}(t+1) &= s_{m-2}(t) \\
&\vdots \\
s_2(t+1) &= s_1(t) \\
s_1(t+1) &= g_m s_m(t) + g_{m-1} s_{m-1}(t) + \cdots + g_2 s_2(t) + g_1 s_1(t)
\end{aligned} \tag{27}
$$

여기서 $s_i(t)$는 t번째 펄스 시의 LFSR의 i번째 값이다. 式 (27)을 行列로 表現하면

$$
\begin{bmatrix} s_m(t+1) \\ s_{m-1}(t+1) \\ \vdots \\ s_2(t+1) \\ s_1(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & & \vdots & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ g_m & g_{m-1} & g_{m-2} & g_{m-3} & \cdots & g_2 & g_1 \end{bmatrix} \begin{bmatrix} s_m(t) \\ s_{m-1}(t) \\ \vdots \\ s_2(t) \\ s_1(t) \end{bmatrix} \tag{28}
$$

또는

$$\mathbf{S}_{t+1} = \mathbf{T} \cdot \mathbf{S}_t \tag{29}$$

가 된다. 여기서 m×m 行列 **T**를 탭 生成 行列 또는 特性 行列(characteristic matrix)이라 한다.

[例題 8] 例題 7에서 다루었던 그림 12를 다시 생각해 보자. LFSR을 구동시킬 初期 씨드 벡터를 $\mathbf{S} = (s_1, s_2, s_3) = (100)$이라 假定하면 클럭에 의해 오른쪽으로 置換을 수행하여 表 2와 같은 레지스터 狀態를 얻을 수 있다. 表 2에서 레지스터 內容狀態의 가장 오른쪽 列은 한 週期 동안의 키 비트를 나타낸다. 置換 레지스터의 週期는 m=3일 경우 $p = 2^m - 1 = 7$임을 주목하자. 따라서 狀態 遷移를 나타내는 그림은 그림 13과 같다.

[例題 9] 式 (28)을 活用하여 다른 접근 方法에 의해 例題 8을 다시 생각해 보자. 式 (28)에 依해 3×3 탭 行列은

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

가 되고

표 6. 오른쪽 置換에 의한 레지스터 內容

置換 番號	레지스터 內容		
i	s_1	s_2	s_3
0	1	0	0(초기값)
1	1	1	0
2	1	1	1
3	0	1	1
4	1	0	1
5	0	1	0
6	0	0	1

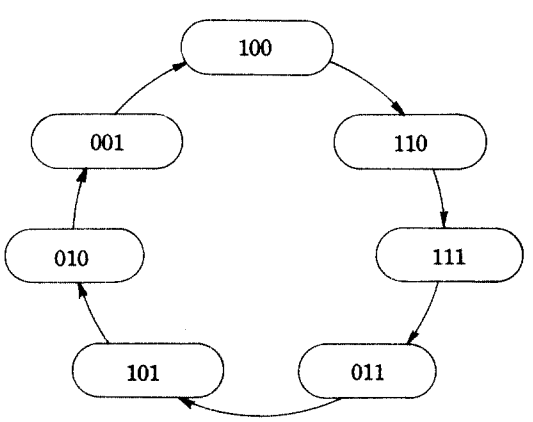

그림 13. LFSR의 裝置動作으로 創出된 狀態遷移圖

$$S_1=\begin{bmatrix}s_3\\s_2\\s_1\end{bmatrix}=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

이므로 式 (29)를 使用하면

$$S_2=T\cdot S_1=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\1\\1\end{bmatrix}=\begin{bmatrix}s_3\\s_2\\s_1\end{bmatrix}$$

가 된다. 繼續해서 反復 遂行하면 다음과 같이 된다.

$$S_3=T\cdot S_2=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}0\\1\\1\end{bmatrix}=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

$$S_4=T\cdot S_3=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix}=\begin{bmatrix}1\\1\\0\end{bmatrix}$$

$$S_5=T\cdot S_4=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}=\begin{bmatrix}1\\0\\1\end{bmatrix}$$

$$S_6=T\cdot S_5=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\1\\0\end{bmatrix}$$

$$S_7=T\cdot S_6=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}=\begin{bmatrix}1\\0\\0\end{bmatrix}$$

← 주기의 끝

$$S_8=T\cdot S_7=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\1\end{bmatrix}=S_1$$

$$S_9=T\cdot S_8=\begin{bmatrix}0&1&0\\0&0&1\\1&0&1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\1\\1\end{bmatrix}=S_1$$

$\vdots$ $\vdots$

S_7 이후의 레지스터 狀態(또는 內容)는 週期的으로 反復됨을 알 수 있다. 따라서 이 結果는 아래처럼 要約될 수 있다.

t	1	2	3	4	5	6	7	8	9	10	…
s_3	0	0	1	1	1	0	1	0	0	1	…
s_2	0	1	1	1	0	1	0	0	1	1	…
s_1	1	1	1	0	1	0	0	1	1	1	…
	← 한 주기 →						← 반복 →				

키 비트 數列 **K**는 위 表의 첫번째 行을 表示한다. 즉,

K=(0011101 0011101 0011101…)

이다. 만일 $T(x)=g_0+g_1x+g_2x^2+\cdots+g_mx^m$가 次數 m인 탭 多項式이라면 $T^{-1}(x)$는 LFSR에서 生成된 계열일 것이다. 이 명제를 證明하기 위해 다음 例題를 살펴보자.

[例題 10] $T(x)=1+x+x^3$을 탭 多項式이라 하면 $T^{-1}(x)=1/T(x)$는 1을 T(x)로 나누는 나눗셈에 依해

$$T^{-1}(x)=1+x+x^2+x^4+x^7+x^8+x^9+x^{11}+x^{14}+x^{15}+x^{16}+x^{18}+x^{21}+\cdots$$

가 되고

$$T^{-1}=K=(1110100\ 1110100\ 1110100\ 1\ \cdots)$$

는 亦是 7비트의 週期를 가진 키 비트 數列이며 위 表의 세번째 行을 表示한다.

一般的으로 $T(x)=g_0+g_1x+g_2x^2+\cdots+g_{m-1}x^{m-1}$가 탭 多項式이고 $S(x)=s_0+s_1x+s_2x^2+\cdots+s_{m-1}x^{m-1}$가 LFSR의 初期 씨드 多項式이라면 出力(키) 多項式 $K(x)=k_0+k_1x+k_2x^2+\cdots+k_{m-1}x^{m-1}$는 다음과 같이 表現될 수 있다.

$$S(x)=K(x)T(x) \qquad (30)$$

여기서

$$K(x)T(x)=k_0g_0+(k_0g_1+k_1g_0)x+(k_0g_2+k_1g_1+k_0s_2)x^2+\cdots+(k_0g_{m-1}+k_1g_{m-2}+\cdots+k_{m-1}g_0)x^{m-1} \qquad (31)$$

이다. 따라서 式 (30)과 (31)로부터 이들 多項式의

係數는 아래와 같은 關係를 갖는다.

$$\begin{aligned} s_0 &= k_0 g_0 \\ s_1 &= k_0 g_1 + k_1 g_0 \\ &\vdots \\ s_{m-1} &= k_0 g_{m-1} + k_1 g_{m-2} + \cdots + k_{m-1} g_0 \end{aligned} \qquad (32)$$

또는

$$s_i = \sum_{i=0}^{i} k_i g_{i-1}, \quad 0 \leq i \leq m-1 \qquad (33)$$

式 (32)의 行列 表現은

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{m-1} \end{bmatrix} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 & 0 \\ g_1 & g_0 & 0 & \cdots & 0 & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 & 0 \\ & \vdots & & \vdots & & \vdots \\ g_{m-1} & g_{m-2} & g_{m-3} & \cdots & g_1 & g_0 \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \\ k_{m-1} \end{bmatrix} \qquad (34)$$

와 같이 될 수 있다. 그러므로 $\mathbf{T}=(g_0, g_1, \cdots, g_{m-1})$와 $\mathbf{K}=(k_0, k_1, \cdots, k_{m-1})$가 주어지면 세번째 벡터 $\mathbf{S}=(s_0, s_1, \cdots, s_{m-1})$는 式 (34)에 依해 쉽게 決定된다.

[例題 11] 탭 多項式은 $T(x)=1+x^2+x^5$, 初期 狀態 벡터를 $\mathbf{K}=(10000)$이라고 가정하자. 그러면 式 (34)에 의해 出力키 벡터 $\mathbf{K}$를 決定할 수 있다. 탭 係數가 $g_0=1$, $g_1=0$, $g_2=1$, $g_3=0$, $g_4=0$, $g_5=1$ 이므로 式 (34)는

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

으로 되어 出力은 $S(x)=1+x^2$, 또는 $\mathbf{S}=(10100)$가 된다.

$S(x)$는 계열의 시작위치를 결정하는 다항식이고, $K(x)$의 첫 m 비트는 LFSR의 초기내용이다. 이 때, $K(x)$는 式(30)으로부터 다음과 같이 표현될 수 있다.

$$K(x) = S(x)T^{-1}(x) \qquad (35)$$

$S(x)=1+x^2$이고 $T(x)=1+x^2+x^5$이므로 式 (35)를 利用하면 $K(x)=1+x^5+x^7+x^9+x^{10}+\cdots$ 또는 $\mathbf{K}=(10000101011\cdots)=(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7, \cdots)$를 얻을 수 있다. 그러므로 $\mathbf{K}$에서 밑줄친(underline) 처음 다섯 디지트는 豫想했던 대로 初期 狀態 $\mathbf{K}=(10000)$을 나타낸다. 初期 씨이드 다항식이 $S(x)=1+x^2$이므로 $x^kS(x)$(法 $1+x^2+x^5$)에 대응되는 출력, $K^k(x)$는 $K(x)$를 오른쪽으로 k 디지트 巡廻置換한 것을 나타낸다. $k=2$에 대해, $x^2S(x)=x^2+x^4$가 된다. 그러면 置換된 內容은 $K^{(2)}(x)=(x^2+x^4)/(1+x^2+x^5)=x^2+x^7+x^9+x^{11}+\cdots$이다.

그러므로 $k=17$번 置換했을 때의 出力은 $x^{17}S(x)=x^{17}+x^{19}$이다. $GF(2^5)$으로부터 두 有限體 元素 $x^{17}=1+x+x^4$과 $x^{19}=x+x^2$을 $x^{17}S(x)$에 代入하면 $x^{17}S(x)=1+x^2+x^4$이 된다. 따라서 $K^{(17)}(x)=(1+x^2+x^4)/(1+x^2+x^5)=1+x^4+x^5+x^6+\cdots$이 된다. 그러므로 17번째 置換했을때 레지스터의 內容은 $K^{(17)}=(10001)$이 됨을 알 수 있다.

이 節에서는 키 비트 數列이 平文과는 獨立的으로 발생되는 回期 逐字 暗號에 관해 알아 보았다. 暗號文 傳送中 어느 한 비트의 損失이 생기는 경우, 더 이상의 傳送이 있기 전에 送受信兩端의 키 發生器를 잃어버리는 결과를 招來하여 損失 이후의 모든 暗號文은 잘못된 平文으로 復元될 것이다. 또한 回期 逐字 暗號는 키 비트 數列이 週期的으로 復元되면 暗號 알고리즘은 깨어질 수 있다. 이런 理由로 会期 逐字 暗號는 화일(file)과 데이타 베이스(database)의 暗號化에 制限的으로 適用될 수 밖에 없다. 그러나 同一한 平文 블럭이 다른 형태의 키 數列로 暗號化되기 때문에 同期 逐字 暗號는

敵의 探索으로 暗號文이 유출되는 것을 막을 수 있다. 또한 暗號文의 插入이나 削除는 同期化를 喪失하는 原因이 되기 때문에 잘못된 暗號文의 插入, 再生, 그리고 削除로 부터 보호할 수 있다.

同期 逐字 暗號는 誤謬 傳播(error propagation)를 일으키지 않는 長點을 가지고 있다. 事實 한 비트의 電送誤謬는 이후의 계속되는 비트들에는 影響을 주지 않을 것이다. 그러나 敵에게는 한 블럭의 비트들을 修正하는 것 보다는 暗號文내의 비트 하나를 修正하는 것이 더 쉽기 때문에 이러한 事實이 短點이 될 수도 있다.

3. 自己 同期 逐字 暗號 시스템

自己 同期 暗號 시스템은 逐字 暗號의 또 다른 種類이며 暗號文 歸還 自動 키 暗號와 平文 歸還 自動키 暗號로 區分된다. 暗號文 自動 키 暗號는 키가 暗號文으로부터 유도되는 고로 暗號文 歸還으로 구성된 自己 同期 逐字 暗號라고 부른다. 各各의 키 비트들은 선행 暗號文 비트들로부터 만들어지기 때문에 驅動 키(priming key)와 平文의 모든 선행 비트를 합한 것에 函數的으로 關聯되어 있다. 이러한 性質은 平文의 統計的 特性이 暗號文으로 되면서 擴散되기 때문에 暗號解讀을 더 어렵게 만들 것이다. 다른 自動 키 암호는 平文 歸還을 갖는 自己 同期 시스템이다. 平文 自動 키 暗號는 키가 暗號化되는 平文으로부터 誘導되는 것이다. 이들 두 自動 키 暗號는 다음 節에서 자세히 다루기로 한다.

3.1 暗號文 歸還 自動 키 暗號 시스템

暗號文 自動 키 暗號는 그림 14와 같은 暗號文 歸還 暗號 시스템을 의미한다. 簡略하게 問題를 풀기 爲하여 그림 14에서 m=4인 경우를 생각해

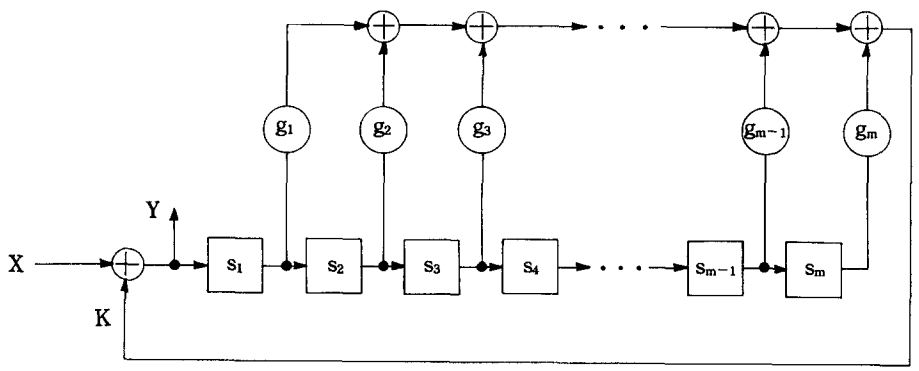

그림 14. 暗號文 歸還 暗號化 裝置

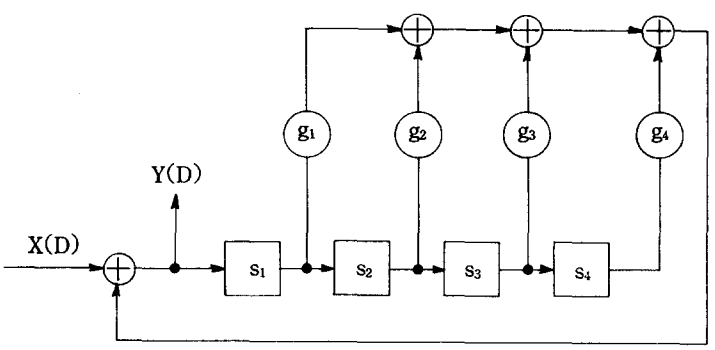

그림 15. 4 段 LFSR로 構成된 暗號文 歸還 暗號化 裝置

보자. 그러면 그림 15에 설계된 것과 같이 4段 LFSR로 이루어진 暗號 시스템이 될 것이다.

T(x)를 遷移 多項式, S(x)를 씨드 多項式이라고 하면 이들은 다음과 같이 遲延 演算子 D로 表現된다.

$$T(D)=g_1+g_2D+g_3D^2+g_4D^3 \qquad (36)$$

$$S(D)=s_1+s_2D^{-1}+s_3D^{-2}+s_4D^{-3} \qquad (37)$$

따라서 暗號文 多項式 y(x)는

$$\begin{aligned}Y(D)&=X(D)+T(D)\cdot D\cdot Y(D)+T(D)S(D)\\&=X(D)+g_1DY(D)+g_2D^2Y(D)+g_3D^3Y(D)+\ g_4D^4Y(D)+(g_1+g_2D+g_3D^2+g_4D^3)\\&\qquad\qquad (s_1+s_2D^{-1}+s_3D^{-2}+s_4D^{-3})\end{aligned} \qquad (38)$$

와 같이 된다. 여기서 $X(D)=x_0+x_1D+x_2D^2+\cdots$는 平文 多項式이고 $Y(D)=y_0+y_1D+y_2D^2+\cdots$는 暗號文 多項式을 나타낸다. 式 (38)을 展開하여 D에 대한 오름次順으로 정돈하면 暗號文 비트 y_i, ($0\leq i\leq 4$, $4\leq i$)는

$$\begin{aligned}
y_0&=x_0+g_1s_1+g_2s_2+g_3s_3+g_4s_4=x_0+\sum_{t=1}^{4}g_ts_t\\
y_1&=x_1+g_1y_0+g_2s_1+g_3s_2+g_4s_3=x_1+g_1y_0+\sum_{t=2}^{4}g_ts_{t-1}\\
y_2&=x_2+g_1y_1+g_2y_0+g_3s_1+g_4s_2\\
&=x_2+\sum_{t=1}^{2}g_ty_{2-t}+\sum_{t=3}^{4}g_ts_{t-2}\\
y_3&=x_3+g_1y_2+g_2y_1+g_3y_0+g_4s_1\\
&=x_3+\sum_{t=1}^{3}g_ty_{3-t}+g_4s_1\\
y_4&=x_4+g_1y_1+g_2y_2+g_3y_3+g_4y_4\\
&=x_4+\sum_{t=1}^{4}g_ty_{4-t}\\
y_5&=x_5+g_1y_4+g_2y_3+g_3y_2+g_4y_1\\
&=x_5+\sum_{t=1}^{4}g_ty_{5-t}\\
y_6&=x_6+g_1y_5+g_2y_4+g_3y_3+g_4y_2\\
&=x_6+\sum_{t=1}^{4}g_ty_{6-t}\\
y_7&=x_7+g_1y_6+g_2y_5+g_3y_4+g_4y_3\\
&=x_7+\sum_{t=1}^{4}g_ty_{7-t}
\end{aligned} \qquad (39)$$

와 같다. 式 (39)의 一般的 表現은 任意의 單位時間 i에서의 暗號文 비트로 다음과 같이 簡略하게 表記된다.

$$y_i=\begin{cases}x_i+\sum_{t=1}^{i}g_ty_{i-t}+\sum_{t=i+1}^{m}g_ts_{t-i}, & 0\leq i\leq m-1\\ x_i+\sum_{t=1}^{m}g_ty_{i-t}, & i\geq m\end{cases} \qquad (40)$$

여기서 m은 LFSR의 段數를 의미한다. 式 (39)를 行列形態로 나타내면 다음과 같다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ y_0 & 0 & 0 & 0 \\ y_1 & y_0 & 0 & 0 \\ y_2 & y_1 & y_0 & 0 \\ y_3 & y_2 & y_1 & y_0 \\ y_4 & y_3 & y_2 & y_1 \\ y_5 & y_4 & y_3 & y_2 \\ y_6 & y_5 & y_4 & y_3 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} + \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ 0 & s_1 & s_2 & s_3 \\ 0 & 0 & s_1 & s_2 \\ 0 & 0 & 0 & s_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \quad (41)$$

式 (41)을 보면 暗號文은 세 項, 즉, 平文項, 暗號文 歸還으로 發生된 項, 그리고 初期 씨드 벡터에 依해 발생된 項 등으로 이루어져 있다. 물론 式 (41)은 式 (40)과 같고 좀더 簡略히 쓰면

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ y_0 & s_1 & s_2 & s_3 \\ y_1 & y_0 & s_1 & s_2 \\ y_2 & y_1 & y_0 & s_1 \\ y_3 & y_2 & y_1 & y_0 \\ y_4 & y_3 & y_2 & y_1 \\ y_5 & y_4 & y_3 & y_2 \\ y_6 & y_5 & y_4 & y_3 \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \quad (42)$$

이 된다. 이것이 暗號化를 위한 行列시스템이다.

[例題 12] 그림 16과 같이 탭 係數가 $\mathbf{T}=(g_1, g_2, g_3, g_4)=(0101)$이고 m=4인 暗號文 歸還 自動키 시스템을 생각해 보자. 2m=8이므로 8비트 平文 벡터는 $\mathbf{X}=(11101001)$이고, 初期 씨드 벡터는 $\mathbf{S}=(s_1, s_2, s_3, s_4)=(0011)$로 LFSR에 저장되어 있다고 假定한다. 그러면 式 (40)을 利用하여 計算되는 8비트 暗號文은

$$y_0 = x_0 + \sum_{t=1}^{4} g_t s_t = x_0 + g_2 s_2 + g_4 s_4$$

$$= x_0 + s_2 + s_4 = 1+0+1=0$$

$$y_1 = x_1 + g_1 y_0 + \sum_{t=2}^{4} g_t s_{t-1} = x_1 + g_2 s_1 + g_4 s_3$$

$$= x_1 + s_1 + s_4 = 1+0+1=0$$

$$y_2 = x_2 + \sum_{t=1}^{2} g_t y_{2-t} + \sum_{t=3}^{4} g_t s_{t-2}$$

$$= x_2 + g_2 y_0 + s_2 = 1+0+0=1$$

$$y_3 = x_3 + \sum_{t=1}^{3} g_t y_{3-t} + g_4 s_1$$

$$= x_3 + y_1 + s_1 = 0+0+0=0$$

$$y_4 = x_4 + \sum_{t=1}^{4} g_t y_{4-t} = x_4 + y_2 + y_0$$

$$= 1+1+0=0$$

$$y_5 = x_5 + \sum_{t=1}^{4} g_t y_{5-t} = x_5 + y_3 + y_1$$

$$= 0+0+0=0$$

$$y_6 = x_6 + \sum_{t=1}^{4} g_t y_{6-t} = x_6 + y_4 + y_2$$

$$= 0+0+1=1$$

$$y_7 = x_7+\sum_{i=1}^{4} g_i y_{7-i} = x_7+y_5+y_3$$

$$= 1+0+0=1$$

가 된다. 그러므로, 8 비트 暗號文은

$$\mathbf{Y} = (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$$
$$= (00100011) \qquad (43)$$

가 된다. 式 (40) 대신 式 (42)를 利用하면 暗號文 歸還을 갖는 暗號文 비트들은 다음과 같이 쉽게 얻을 수 있다.

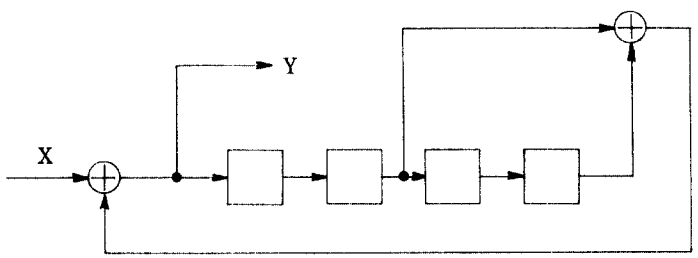

그림 16. m=4이고 T=(0101)인 暗號文 歸還 暗號化 裝置

[例題 13] 式 (42)의 行列 表現式을 利用해서 例題 12를 다시 풀어보자. 2m 平文 系列을 $\mathbf{X}=(11101001)$, 初期 씨드 벡터를 $\mathbf{S}=(0011)$, 그리고 傳達係數를 $\mathbf{T}=(0101)$라고 假定했기 때문에 暗號文 비트를 計算하기 위한 行列 시스템을 다음과 같이 즉시 構成할 수 있다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 1 \\ y_0 & 0 & 0 & 1 \\ y_1 & y_0 & 0 & 0 \\ y_2 & y_1 & y_0 & 0 \\ y_3 & y_2 & y_1 & y_0 \\ y_4 & y_3 & y_2 & y_1 \\ y_5 & y_4 & y_3 & y_2 \\ y_6 & y_5 & y_4 & y_3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+1 \\ 1+1 \\ 1+y_0 \\ 0+y_1 \\ 1+y_2+y_0 \\ 0+y_3+y_1 \\ 0+y_4+y_2 \\ 1+y_5+y_3 \end{bmatrix}$$

따라서,

$y_0=1+1=0$ $\quad y_4=1+y_2+y_0=1+1+0=0$

$y_1=1+1=0$ $\quad y_5=0+y_3+y_1=0+0+0=0$

$y_2=1+y_0=1+0=1$ $\quad y_6=0+y_4+y_2=0+0+1=1$

$y_3=0+y_1+0+0=0$ $\quad y_7=1+y_5+y_3=1+0+0=1$

가 되고 暗號文 비트는,

$$\mathbf{Y}=(00100011) \qquad (44)$$

와 같이 生成되고 이는 式 (43)과 同一하다.

例題 12와 13에서 보았듯이 式 (41)은 다음과 같이 그 表現을 擴大시킬 수 있다.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{2m-2} \\ y_{2m-1} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{2m-2} \\ x_{2m-1} \end{bmatrix} \begin{bmatrix} s_1 & s_2 & s_3 & \cdots & s_{m-1} & s_m \\ y_0 & s_1 & s_2 & \cdots & s_{m-2} & s_{m-1} \\ y_1 & y_0 & s_1 & \cdots & s_{m-3} & s_{m-2} \\ & \vdots & & & \vdots & \\ y_{2m-3} & y_{2m-4} & y_{2m-5} & \cdots & y_{m-1} & y_{m-2} \\ y_{2m-2} & y_{2m-3} & y_{2m-4} & \cdots & y_{m-2} & y_{m-1} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} \quad (45)$$

$\mathbf{x}_i$를 任意의 單位時間 i에서의 平文 系列이라 하고 $\mathbf{y}_i$를 그에 對應되는 暗號文 系列이라고 하면 $\mathbf{x}_i=(x_i, x_{i+1}, \cdots, x_{i+m-1})$, $\mathbf{y}_i=(y_i, y_{i+1}, \cdots, y_{i+m-1})$로 表現된다. 自動 키 暗號에서 LFSR의 初期 狀態는 처음 m클럭 펄스로 初期値가 빠져나간 이후에는 出力에 無關해 진다. 그러므로 式 (45)는 다음과 같이 修正될 수 있다.

$$\mathbf{y}_{i+m}=\mathbf{x}_{i+m}+\mathbf{T}\mathbf{Y}(i), \quad 0\leq i \quad (46)$$

여기서

$$\mathbf{Y}(i)=\begin{bmatrix} y_{i+m-1} & y_{i+m} & \cdots & y_{i+2m-3} & y_{i+2m-2} \\ y_{i+m-2} & y_{i+m-1} & \cdots & y_{i+2m-4} & y_{i+2m-3} \\ \vdots & & & \vdots & \\ y_{i+1} & y_{i+2} & \cdots & y_{i+m-1} & y_{i+m} \\ y_i & y_{i+1} & \cdots & y_{i+m-2} & y_{i+m-1} \end{bmatrix} \quad (47)$$

이고 式 (46)으로부터 탭 係數 $\mathbf{T}=(g_1, g_2, \cdots, g_m)$는 다음 節次에 依해 決定될 수 있다.

$$\mathbf{T}=(\mathbf{y}_{i+m}+\mathbf{x}_{i+m})\mathbf{Y}^{-1}(i) \quad (48)$$

i=0일때 式 (48)은

$$\mathbf{T}=(\mathbf{y}_m+\mathbf{x}_m)\mathbf{Y}^{-1}(0) \quad (49)$$

가 된다.

[例題 14] 平文은 $\mathbf{X}=(11101001)$이고 그에 對應하는 暗號文이 $\mathbf{Y}=(00100011)$일 때 그림 16을 다시 생각해 보자. 탭 係數 T와 初期 씨드 S를 결정해 보자. i=0와 m=4에 대해 式 (47)은 다음과 같이 된다.

$$\mathbf{Y}(0)=\begin{bmatrix} y_3 & y_4 & y_5 & y_6 \\ y_2 & y_3 & y_4 & y_5 \\ y_1 & y_2 & y_3 & y_4 \\ y_0 & y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

m=4에 對한 式 (49)를 利用하면

$$\mathbf{T}=(\mathbf{y}_4+\mathbf{x}_4)\mathbf{Y}^{-1}(0)=(0011+1001)\begin{bmatrix} 0100 \\ 0010 \\ 0001 \\ 0000 \end{bmatrix}=(0101)$$

이다. 따라서 탭 係數(歸還係數 또는 傳達係數)는 완전하게 決定되었다. i=0에서 系列이 始作된다는 條件하에서 式 (40)은 初期 씨드를 決定하는 데 사용될 수 있다. 즉,

$$y_0=x_0+s_2+s_4$$
$$y_1=x_1+s_1+s_3$$
$$y_2=x_2+y_0+s_2$$
$$y_3=x_3+y_1+s_1$$

가 되어

$$s_1=y_3+x_3+y_1=0+0+0=0$$
$$s_2=y_2+x_2+y_0=1+1+0=0$$
$$s_3=y_1+x_1+s_1=0+1+0=1$$
$$s_4=y_0+x_0+s_2=0+1+0=1$$

가 얻어진다. 그러므로 初期 씨드 벡터는

$$\mathbf{S}=(s_1, s_2, s_3, s_4)=(0011)$$

이다.

마지막으로 그림 17에 나타낸 暗號文 歸還에 依한 復號 시스템을 생각해 보자. 暗號化 시스템에서 使用했던 똑같은 T와 S를 가지고 다음 關係式을 利用해서 平文을 再生해보자.

$$x_i=\begin{cases} y_i+\sum_{t=1}^{i} g_t y_{i-t}+\sum_{t=i+1}^{m} g_t s_{t-i}, & 0\leq i\leq m-1 \\ y_i+\sum_{t=1}^{m} g_t y_{i-t}, & i\geq m \end{cases} \quad (50)$$

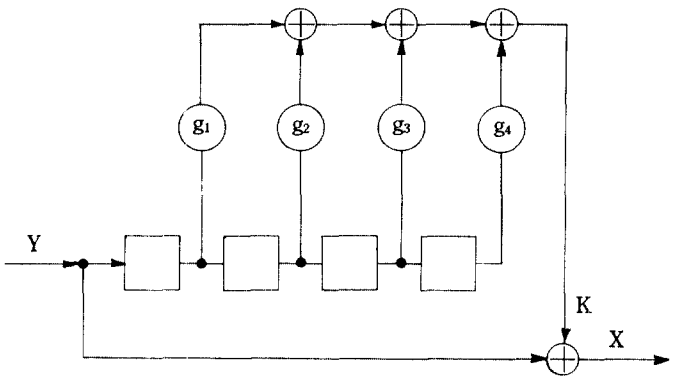

그림 17. 4段 LFSR을 利用한 暗號文 復號化 裝置

[例題 15] 暗號文 $\mathbf{Y}=(00100011)$에 對應되는 平文 $\mathbf{X}$를 復元하기 위한 復號裝置인 그림 17을 생각해 보자. 이때 앞에서 假定했던 것과 같이 탭 係數는 $\mathbf{T}=(0101)$이고 初期 씨드 벡터는 $\mathbf{S}=(0011)$이다. 式 (50)을 利用하여 復元하고자 하는 平文을 決定해 보자.

$$x_0=y_0+\sum_{t=1}^{4}g_ts_t=y_0+s_2+s_4$$
$$=0+0+1=0$$
$$x_1=y_1+g_1y_0+\sum_{t=2}^{4}g_ts_{t-1}=y_1+s_1+s_3$$
$$=0+0+1=1$$
$$x_2=y_2+\sum_{t=1}^{2}g_ty_{2-t}+\sum_{t=3}^{4}g_ts_{t-2}$$
$$=y_2+y_0+s_2+1+0+0=1$$
$$x_3=y_3+\sum_{t=1}^{3}g_ty_{3-t}+g_4s_1$$
$$=y_3+y_1+s_1=0+0+0=0$$
$$x_4=y_4+\sum_{t=1}^{4}g_ty_{4-t}=y_4+y_2+y_0$$
$$=0+1+0=1$$
$$x_5=y_5+\sum_{t=1}^{4}g_ty_{5-t}=y_5+y_3+y_1$$
$$=0+0+0=0$$
$$x_6=y_6+\sum_{t=1}^{4}g_ty_{6-t}=y_6+y_4+y_2$$
$$=1+0+1=0$$
$$x_7=y_7+\sum_{t=1}^{4}g_ty_{7-t}=y_7+y_5+y_3$$
$$=1+0+0=1$$

그러므로 復元된 平文은 豫想한 바와 같이

$$\mathbf{X}=(11101001)$$

이다.

3.2 平文 歸還 自動 키 暗號

平文 歸還 自動 키 暗號(plaintext autokey cipher)는 置換 레지스터의 入力이 平文인 自己 同期 暗號 시스템(self-syncronizing cipher system)을 말한다.

時間 i에서의 平文 系列을 $\mathbf{x}_i=(x_i, x_{i+1}, \cdots, x_{i+m-1})$라 하고 이에 對應하는 暗號 系列을 $\mathbf{y}_i=(y_i, y_{i+1}, \cdots, y_{i+m-1})$라 하면

$$\mathbf{y}_{i+m}=\mathbf{x}_{i+m}+\mathbf{X}(i)\mathbf{T}\ ,\ 0\leq i \qquad (51)$$

여기서

$$\mathbf{X}(i)=\begin{bmatrix} x_{i+m-1} & x_{i+m} & \cdots & x_{i+2m-2} \\ x_{i+m-2} & x_{i+m-1} & \cdots & x_{i+2m-3} \\ \vdots & & & \vdots \\ x_i & x_{i+1} & \cdots & x_{i+m-1} \end{bmatrix}$$

의 關係를 갖고 式 (51)로부터 置換 레지스터의 탭 係數(tap coefficient) $\mathbf{T}$는

$$\mathbf{T}=\mathbf{X}^{-1}(i)(\mathbf{y}_{i+m}+\mathbf{x}_{i+m}) \qquad (52)$$

로 얻어진다.

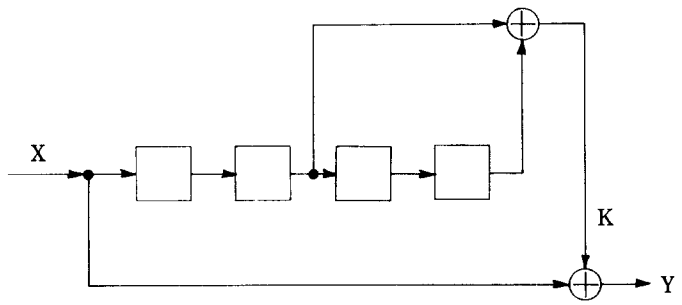

그림 18. 平文 入力을 갖는 自己 同期 暗號化 裝置

[例題 16] 그림 18과 같이 置換레지스터의 入力이 平文인 自己 同期 暗號 시스템에 대해 생각해 보자. 탭 係數는 $\mathbf{T}=(g_1, g_2, g_3, g_4)=(0101)$, 初期 狀態는 $\mathbf{S}=(s_1, s_2, s_3, s_4)=(0011)$이고 平文 系列은 $\mathbf{X}=(11101001)$라 假定하자. m=4이므로 y_0에서 y_7까지의 暗號文 비트는 式 (40)으로부터 y_{i-t}를 x_{i-t}로 바꿈으로써 다음과 같이 구할 수 있다.

$$y_0=x_0+\sum_{t=1}^{4} g_t s_t=x_0+s_2+s_4$$

$$=1+0+1=0$$

$$y_1=x_1+g_1x_0+\sum_{t=2}^{4} g_t s_{t-1}=x_1+s_1+s_3$$

$$=1+0+1=0$$

$$y_2=x_2+\sum_{t=1}^{2} g_t x_{2-t}+\sum_{t=3}^{4} g_t s_{t-2}$$

$$=x_2+x_0+s_2=1+1+0=0$$

$$y_3=x_3+\sum_{t=1}^{3} g_t y_{3-t}+g_4 s_1$$

$$=x_3+x_1+s_1=0+1+0=1$$

$$y_4=x_4+\sum_{t=1}^{4} g_t y_{4-t}=x_4+x_2+x_0$$

$$=1+1+1=1$$

$$y_5=x_5+\sum_{t=1}^{4} g_t y_{5-t}=x_5+x_3+x_1$$

$$=0+0+1=1$$

$$y_6=x_6+\sum_{t=1}^{4} g_t x_{6-t}=x_6+x_4+x_2$$

$$=0+1+1=0$$

$$y_7=x_7+\sum_{t=1}^{4} g_t x_{7-t}=x_7+x_5+x_3$$

$$=1+0+0=1$$

따라서 暗號文 系列은 $\mathbf{Y}=(00011101)$가 된다.

[例題 17] 그림 18을 다시 利用하여 탭 벡터 **T**와 씨드 벡터 **S**를 결정하는 문제에 대해 생각해 보자. 平文 $\mathbf{X}=(11101001)$에 對應되는 暗號文 $\mathbf{Y}=(00011101)$은 例題 16에서 보듯이 이미 求하였다. i=0, m=4일 때 式 (52)는

$$\mathbf{T}=\mathbf{X}^{-1}(0)(y_4+x_4)$$

가 되며 $\mathbf{X}(0)$는

$$\mathbf{X}(0)=\begin{bmatrix} x_3 x_4 x_5 x_6 \\ x_2 x_3 x_4 x_5 \\ x_1 x_2 x_3 x_4 \\ x_0 x_1 x_2 x_3 \end{bmatrix}=\begin{bmatrix} 0\,1\,0\,0 \\ 1\,0\,1\,0 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{bmatrix}$$

이다. $\mathbf{X}(0)$의 行列式은 $|\mathbf{X}(0)|=0$이므로 行列 $\mathbf{X}(0)$는 특이해이다. 따라서 $\mathbf{X}^{-1}(0)$는 결정할 수 없다. 결국 이 경우 탭 係數 **T**를 決定하는 것은 不可能하다. 이 理由는 平文에 利用되는 歸還 經路가 없기 때문이다.

[例題 18] 그림 19와 같이 平文이 歸還되는 自動키 復號에 대해 생각해 보자.

例題 16에서 分析한 것 처럼 $\mathbf{T}=(0101)$, $\mathbf{S}=(0011)$, 그리고 $\mathbf{Y}=(00011101)$로 하자. y_{i-t}를 x_{i-t}로 代置한 式 (50)을 利用하면 平文 비트는 아래와 같이 復號될 수 있다.

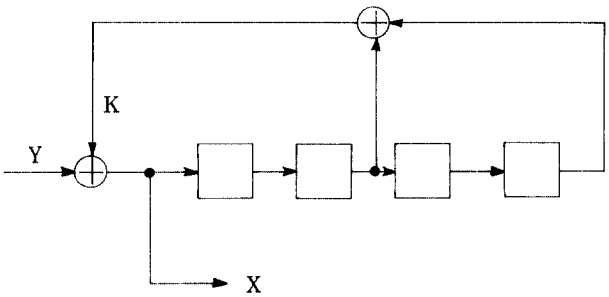

그림 19. 平文 歸還 復號化 裝置

$x_0=y_0+\sum_{t=1}^{4} g_t s_t=y_0+s_2+s_4$

$=0+0+1=0$

$x_1=y_1+g_1x_0+\sum_{t=2}^{4} g_t s_{t-1}=y_1+s_1+s_3$

$=0+0+1=1$

$x_2=y_2+\sum_{t=1}^{2} g_t y_{2-t}+\sum_{t=3}^{4} g_t s_{t-2}$

$=y_2+x_0+s_2=0+1+0=1$

$x_3=y_3+\sum_{t=1}^{3} g_t y_{3-t}+g_4s_1$

$=y_3+x_1+s_1=1+1+0=0$

$x_4=y_4+\sum_{t=1}^{4} g_t y_{4-t}=y_4+x_2+x_0$

$=1+1+1=1$

$x_5=y_5+\sum_{t=1}^{4} g_t y_{5-t}=y_5+x_3+x_1$

$=1+0+1=0$

$x_6=y_6+\sum_{t=1}^{4} g_t y_{6-t}=y_6+x_4+x_2$

$=0+1+1=0$

$x_7=y_7+\sum_{t=1}^{4} g_t y_{7-t}=y_7+x_5+x_3$

$=1+0+0=1$

따라서 復號된 平文은 $\mathbf{x}=(11101001)$이다.

(다음 號에 계속)

□ 著者紹介

李 晩 榮(正會員)

1924年 11月 30日生

서울大學校 電氣工學科 工學士(BSEE)

美國 Colorado大學校 工學碩士(MSEE) 및 工學博士(Ph.D.)

美國 Virginia州立大 工大敎授

美國 California Institute of Technology, JPL 硏究員

國防科學硏究所 第1副所長／韓國電子通信 社長／三星半導體通信社長／漢陽大 副總長／現 漢陽大 名譽敎授／韓國通信情報保護學會 會長

著書： Error Correcting Coding Theory, McGraw-Hill, New York, 1989.

지난 號의 誤記를 아래와 같이 바로 잡습니다.

面 段 行	誤	正
50 左 17	k_{m-1}	k_{2m-1}
18	z_{m-1}	z_{2m-1}
50 右 9	$K=(k_{70}, k_1, \cdots, k_{77})$	$K=(k_0, k_1, \cdots, k_7)$
12	$k_j+s_j=z_{7j}$가 $z_{7j}+k_j=s_j$	$k_j+s_j=z_j$가 $z_j+k_j=s_j$
52 左 12	g_i	g_1
右 20	同日	同一

以上에서 講述한 逐字(stream) 暗號시스템에 관한 大部分의 分析技法은 著者特有의 解釈方式으로서 他 어떤 文獻에서도 찾아볼 수 없는 獨自的인 內容임. 따라서, 讀者는 參考文獻에 明示없이 無斷盜用이나 転用하는 것을 禁해주기 바람.