سوال ۱)

۱.۱

با توجه به این‌که به ازای هر x داریم: $DES_{K_w}(x) = DES_{K_w}^{-1}(x)$

بنابراین عملیات رمزنگاری و رمزگشایی باید یکسان باشد. پس زیرکلیدهای تولید شده باید دارای رابطه‌ی زیر با یکدیگر باشند:

$$k_{i+1} = k_{16-i} \quad , i = 0,1, \dots ,7$$

۱.۲

$$C_{i+1} = C_{16-i} \quad , i = 0,1, \dots ,7$$

$$D_{i+1} = D_{16-i} \quad , i = 0,1, \dots ,7$$

بنابراین چهار کلید ضعیف هنگامی بدست می‌آیند که:

$$C_0 = FFFFFFF_{16} \text{ یا } 0000000_{16}$$

$$and$$

$$D_0 = FFFFFFF_{16} \text{ یا } 0000000_{16}$$

بنابراین چهار کلید ضعیف پس از PC-1 به صورت زیر خواهند بود:

$$(C_0\, , D_0) = 00000000000000_{16}$$

$$(C_0\, , D_0) = 0000000FFFFFFF_{16}$$

$$(C_0\, , D_0) = FFFFFFF0000000_{16}$$

$$(C_0\, , D_0) = FFFFFFFFFFFFFF_{16}$$

۱.۳

احتمال انتخاب یکی از کلیدهای ضعیف برابر است با:

$$\frac{4}{2^{56}} = \frac{2^2}{2^{56}} = 2^{-54}$$

سوال ۲)

۲.۱

$$A \oplus B = (AB')\vee(A'B)$$

$$A' \oplus B' = \big((A')(B')'\big)\vee\big((A')'(B')\big) = (A'B)\vee(AB') = A \oplus B$$

$$A' \oplus B = (A'B')\vee(AB) = (A'\vee A)(A'\vee B)(B'\vee A)(B'\vee B) = (A'\vee B)(B'\vee A) =$$

$$((A'\vee B)' \vee (B'\vee A)')' = \big((AB')\vee(A'B)\big)' = (A \oplus B)'$$

۲.۲

با توجه به این‌که تابع PC - 1 یک جایگشت مشخصی را بر روی بیت‌ها اعمال می‌کند، مکمل کردن (یا اصطلاحا flip کردن) بیت‌ها قبل یا بعد از آن تفاوتی به وجود نمی‌آورد؛ بنابراین $PC-1(k') = \big(PC-1(k)\big)'$ است.

۲.۳

مشابه با قسمت قبل، توابع چرخش زیر کلیدها نیز دارای جایگشت‌های مشخصی هستند و مکمل کردن (یا اصطلاحا flip کردن) بیت‌ها قبل یا بعد از آن تفاوتی ایجاد نمی‌کند؛ پس بنابراین $LS_i(C'_{i-1}) = \big(LS_i(C_{i-1})\big)'$ است.

۲.۴

با توجه به این‌که PC – 2 نیز مانند PC – 1 یک جایگشت خطی است، همه‌ی عملیات‌ها برای تولید کلید خطی هستند؛ بنابراین اگر k' ورودی باشد، کلیدهای تولید شده k'_i هستند.

۲.۵

مشابه با قسمت‌های قبلی، IP نیز خطی بوده و $IP(x') = (IP(x))'$ است.

۲.۶

تابع E که هر بیت در یک بردار را به یک یا دو بیت در یک بردار بزرگتر نگاشت یا اصطلاحا map می‌کند. با توجه به این که هیچ عملیات ترکیبی انجام نمی‌شود، هر بیت در بردار گسترش یافته حاصل از یک بیت در بردار اصلی است، پس مکمل کردن یک بیت قبل یا بعد از نگاشت کردن تفاوتی ایجاد نمی‌کند؛ بنابراین $E(R'_i) = (E(R_i))$ است.

۲.۷

می‌دانیم که $f(R_i\,, k) = P\big(S(E(R_i) \oplus k)\big)$ است، بنابراین طبق نتایجی که تا کنون بدست آورده‌ایم، داریم:

$$f(R'_i\,,k') = P\big(S(E(R'_i) \oplus k')\big) = P\big(S(E(R_i)' \oplus k')\big) = P\big(S(E(R_i) \oplus k')\big) = f(R_i\,,k)$$

با توجه به این‌که $R_i = L_{i-1} \oplus f(R_{i-1}\,,k)$ است، داریم:

$$\Rightarrow \quad L'_{i-1} \oplus f(R'_{i-1}\,,k') = L'_{i-1} \oplus f(R_{i-1}\,,k) = \big(L_{i-1} \oplus f(R_{i-1}\,,k)\big)' = R'_i$$

۲.۸

با در نظر گرفتن نتایج قسمت ۷، در صورتی که بیت‌های ورودی مکمل شوند، خروجی هر دور نیز مکمل می‌شود؛ بنابراین بیت‌های ورودی به تابع IP^{-1} نیز مکمل شده‌اند و با توجه به این‌که تابع IP^{-1} خطی است، پس $IP^{-1}(x') = (IP^{-1}(x))'$ است. بنابراین نتیجه مورد نظر حاصل می‌شود:

$$y = DES_k(x) \;\; \Rightarrow \;\; y' = DES_{k'}(x')$$

سوال ۳)

۳.۱

در مرحله اول، $IP(x)$ بیت ۵۷ ام را به بیت ۳۳ ام نگاشت می‌دهد، بنابراین همه‌ی بیت‌ها برابر با صفر بوده و تنها بیت ۳۳ ام برابر با یک است، پس $L_0 = 0$ و $R_0 = 2^{31}$ است.

هنگام محاسبه $f(R_0)$، می‌توانیم کلید صفر را به دلیل این‌که $a \oplus 0 = a$ است، به حساب نیاوریم. بنابراین $x \oplus k = x$

سپس $E(R_0)$، بیت اول R_0 را به بیت ۲ ام و ۴۸ ام نگاشت می‌کند؛ یعنی S_{2-7} همه در ورودی مقدار 0 می‌گیرند و S_1 در ورودی دارای مقدار 010000_2 بوده و S_8 مقدار 000001_2 را به عنوان ورودی می‌گیرد.

۳.۲

حداقل تعداد بیت‌های خروجی (در هر S-box) که به ازای یک بیت تغییر در ورودی، تغییر خواهند کرد؛ برابر با ۲ است.

۳.۳

با عبور ورودی‌ها از S-box ها، خروجی‌های زیر حاصل می‌شود:

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
ورودی	010000	000000	000000	000000	000000	000000	000000	000001
خروجی	0011	1111	1010	0111	0010	1100	0100	0001

پس از عملیات Permutation بر روی خروجی S-box داریم:

$$1101\ 0000\ 0101\ 1000\ 0101\ 1011\ 1001\ 1110_2$$

سپس این مقدار با L_0، XOR می‌شود تا R_1 را تولید کند ولی با توجه به این‌که L_0 برابر با 0 است، می‌توان از این مرحله صرف نظر کنیم. بنابراین مقادیر R_1 و L_1 به صورت زیر بدست می‌آیند:

$$L_1 = R_0 = 8000\ 0000_{16}$$

$$R_1 = D058\ 5B9E_{16}$$

۳.۴

در حالتی که plaintext کاملا صفر است، با توجه به این‌که همه‌ی بیت‌های ورودی صفر تفاوتی در $IP(x)$ ایجاد نمی‌کنند، بنابراین خروجی به صورت زیر خواهد بود:

$$L_0 = R_0 = 0$$

سپس خروجی S-box ها به صورت زیر حاصل می‌شود:

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
ورودی	000000	000000	000000	000000	000000	000000	000000	000000
خروجی	1110	1111	1010	0111	0010	1100	0100	1101

بعد از عملیات Permutation داخلی تابع f و XOR کردن با L_0 داریم:

$$R_1 = 1101\ 1000\ 1101\ 1000\ 1101\ 1011\ 1011\ 1100_2$$

$$L_1 = 0000\ 0000_{16}$$

اکنون پاسخ‌های بدست آمده از این قسمت و قسمت قبل (۳.۳) را با یکدیگر XOR می‌کنیم تا تعداد بیت‌هایی که دچار تغییر شده‌اند را بدست آوریم:

$$1101\ 0000\ 0101\ 1000\ 0101\ 1011\ 1001\ 1110_2$$

$$\oplus$$

$$1101\ 1000\ 1101\ 1000\ 1101\ 1011\ 1011\ 1100_2$$

$$=$$

$$0000\ 1000\ 1000\ 0000\ 1000\ 0000\ 0010\ 0010_2$$

همان‌طور که مشاهده می‌شود، ۵ بیت از R_1 دچار تغییر شده‌است، همچنین در یک بیت از L_1 نیز تغییر رخ داده است. بنابراین مشاهده می‌شود که به ازای تغییر یک بیت در ورودی، ۶ بیت در خروجی دچار تغییر می‌شوند.

سوال ۴)

۴.۱

$$A(x) + B(x) = (x^2 + 1) + (x^3 + x^2 + 1) \ \ mod\ P(x) = x^3$$

$$A(x) * B(x) = (x^2 + 1) * (x^3 + x^2 + 1)$$

$$= x^5 + x^4 + x^2 + x^3 + x^2 + 1$$

$$= x^5 + x^4 + x^3 + 1$$

حال باید حاصل $A(x) * B(x)\ \ mod\ \ P(x)$ را محاسبه کنیم. با توجه به این‌که $x^4 + x + 1 = 0\ \ mod\ P(x)$ پس می‌توان نوشت:

$$x^4 = x + 1\ \ mod\ P(x)$$

$$\Rightarrow\ \ A(x) * B(x)\ \ \ mod\ \ P(x)$$

$$= x^5 + x^4 + x^3 + 1\ \ \ mod\ P(x)$$

$$= x(x + 1) + (x + 1) + x^3 + 1\ \ \ mod\ P(x)$$

$$= x^2 + x + x + 1 + x^3 + 1\ \ mod\ P(x)$$

$$= x^3 + x^2$$

۴.۲

$$A(x) + B(x) = (x^2 + 1) + (x + 1)\ \ mod\ P(x) = x^2 + x$$

$$A(x) * B(x) = (x^2 + 1) * (x + 1) = x^3 + x^2 + x + 1$$

$$\Rightarrow\ \ A(x) * B(x)\ \ \ mod\ \ P(x) = x^3 + x^2 + x + 1$$

سوال ۵)

۵.۱

$$\forall\ a_i \in GF(2) = \{0,1\} : p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

هدف پیدا کردن $p(x)$ های درجه ۳ است، بنابراین $a_3 = 1$. همچنین باید در نظر داشته باشیم که چند جمله‌ای مورد نظر باید *irreducible* باشد؛ یعنی در $GF(2)$ ریشه نداشته باشد، یعنی $p(0) \neq 0$ و $p(1) \neq 0$

$$p(0) \neq 0\ \ \ \Rightarrow\ \ \ a_0 \neq 0\ \ \ \Rightarrow\ \ \ a_0 = 1$$

(اگر مقدار a_0 برابر با یک نباشد، چند جمله‌ای *irreducible* نیست و می‌تواند از یک x فاکتور گرفته و آن را به دو عبارت با درجه کمتر تبدیل کنیم)

$$p(1) \neq 0 \quad \Rightarrow \quad 1 \times 1 + a_2 + a_1 + 1 \neq 0 \quad \Rightarrow \quad a_2 + a_1 \neq 0 \quad mod\ 2$$

بنابراین ۲ حالت داریم:

$$a_2 = 1 \quad \Rightarrow \quad p(x) = x^3 + x^2 + 1$$

$$a_1 = 1 \quad \Rightarrow \quad p(x) = x^3 + x + 1$$

بنابراین چند جمله‌ای های *irreducible* از درجه ۳ بر روی میدان $GF(2)$ به صورت زیر می‌باشند:

$$x^3 + x^2 + 1$$

$$x^3 + x + 1$$

۵.۲

$$\forall\ a_i \in GF(2) = \{0,1\} : p(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

هدف پیدا کردن $p(x)$ های درجه ۴ است، بنابراین $a_4 = 1$. همچنین باید در نظر داشته باشیم که چند جمله‌ای مورد نظر باید *irreducible* باشد؛ یعنی در $GF(2)$ ریشه نداشته باشد، یعنی $p(0) \neq 0$ و $p(1) \neq 0$

$$p(0) \neq 0 \quad \Rightarrow \quad a_0 \neq 0 \quad \Rightarrow \quad a_0 = 1$$

(اگر مقدار a_0 برابر با یک نباشد، چند جمله‌ای *irreducible* نیست و می‌تواند از یک x فاکتور گرفته و آن را به دو عبارت با درجه کمتر تبدیل کنیم)

$$p(1) \neq 0 \quad \Rightarrow \quad 1 \times 1 + a_3 + a_2 + a_1 + 1 \neq 0 \quad \Rightarrow \quad a_3 + a_2 + a_1 \neq 0 \quad mod\ 2$$

بنابراین ۴ حالت داریم:

(اگر دو تا از ضرایب a_i برابر با یک یا همه آن‌ها برابر با صفر باشند، نامساوی $a_3 + a_2 + a_1 \neq 0 \quad mod\ 2$ برقرار نمی‌شود)

$$a_3 = 1,\ \ a_2 = 1,\ \ a_1 = 1 \quad \Rightarrow \quad p(x) = x^4 + x^3 + x^2 + x + 1$$

$$a_3 = 1 \quad \Rightarrow \quad p(x) = x^4 + x^3 + 1$$

$$a_1 = 1 \quad \Rightarrow \quad p(x) = x^4 + x + 1$$

$$a_2 = 1 \quad \Rightarrow \quad p(x) = x^4 + x^2 + 1 \qquad \times$$

چند جمله‌ای آخر *reducible* است، یعنی داریم:

$$p(x) = x^4 + x^2 + 1 \quad mod\ 2 = (x^2 + x + 1)^2$$

سه چند جمله‌ای دیگر به هیچ کدام از عوامل درجه پایین‌تر خود تجزیه نمی‌شوند و $irreducible$ هستند. بنابراین چند جمله‌ای های $irreducible$ از درجه ۴ بر روی میدان $GF(2)$ به صورت زیر می‌باشند:

$$x^4 + x^3 + x^2 + x + 1$$

$$x^4 + x^3 + 1$$

$$x^4 + x + 1$$

سوال ۶)

با توجه به این‌که همه S-box ها عملکرد یکسانی دارند و ورودی همه آن‌ها برابر با FF_{16} است، با استفاده از جدول S-box ها (جدول ۴.۳ کتاب) می‌توان مقدار خروجی S-box ها را بدست آورد.

Table 4.3 AES S-Box: Substitution values in hexadecimal notation for input byte (*xy*)

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	$\boxed{\text{F}}$
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
$\boxed{\text{F}}$	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	$\boxed{16}$

بنابراین خروجی S-box ها برابر است با:

$$B = ByteSub(A) = \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix}$$

با توجه به این‌که همه درایه‌ها یکسان هستند و جابه‌جایی آن‌ها تفاوتی را ایجاد نمی‌کند، بنابراین می‌توان از عملیات ShiftRow صرف نظر کرد.

برای عملیات MixColumn باید ضرب زیر را در میدان $GF(2^8)$ انجام دهیم:

$$C = MixColumn(B) = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix}$$

$$= \begin{bmatrix} (02+03+01+01) \times 16 \\ (01+02+03+01) \times 16 \\ (01+01+02+03) \times 16 \\ (03+01+01+02) \times 16 \end{bmatrix}$$

در میدان توسعه یافته $GF(2^8)$ عملیات به صورت زیر انجام می‌شود:

$$01 \equiv 0000\ 0001 \ \equiv 1\ ,\ \ 02 \equiv 0000\ 0010 \ \equiv x\ ,\ \ 03 \equiv 0000\ 0011 \ \equiv x+1$$

$$\Rightarrow \quad 01+01+02+03 \equiv 1+1+x+x+1 \equiv 1 \quad mod\ 2$$

$$\Rightarrow \quad 01 \times 16 = 16$$

بنابراین خروجی عملیات MixColumn تغییری نمی‌کند و برابر است با:

$$C = MixColumn(B) = \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix}$$

در نهایت عملیات AddRoundKey به صورت زیر انجام می‌شود:

(کلید دور اول برابر با کلید تغییر نیافته AES می‌باشد، همان کلید تمام یک اولیه)

$$C \oplus K = \begin{bmatrix} 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 \end{bmatrix} \oplus \begin{bmatrix} FF & FF & FF & FF \\ FF & FF & FF & FF \\ FF & FF & FF & FF \\ FF & FF & FF & FF \end{bmatrix}$$

$$= \begin{bmatrix} E9 & E9 & E9 & E9 \\ E9 & E9 & E9 & E9 \\ E9 & E9 & E9 & E9 \\ E9 & E9 & E9 & E9 \end{bmatrix}$$

سوال ۷)

۷.۱

با توجه به این‌که طول کلید از طول بلوک (بر حسب بیت) کمتر است، ممکن است که بر اساس اصل لانه کبوتری، هر کلید به یک متن رمز شده یکتا نگاشت شود؛ ولی این امکان هم وجود دارد که چند کلید به یک متن رمز شده نگاشت شوند.

اگر t را تعداد جفت‌های plaintext و ciphertext مورد استفاده برای شکستن رمز در نظر بگیریم، احتمال پیدا کردن یک کلید مثبت کاذب برابر است با : 2^{k-tn}

بنابراین بسته به مقادیر n و k ، اگر از دو جفت plaintext و ciphertext استفاده کنیم، احتمال بدست آوردن کلید کاذب بسیار کم شده و اطمینان بیشتری بدست می‌آید. در صورتی که آخرین کلید مورد بررسی درست باشد، بدترین حالت، باید 2^k کلید را چک کنیم.

۷.۲

با دانستن بردار اولیه (IV) در مد CBC، شکستن رمز همانند مد ECB شده و تفاوت چندانی ندارد. تنها تفاوت بین این دو مد، وجود XOR در مد CBC است که باید قبل از هر بررسی با i-1 اُمین متن رمزشده یا IV انجام شود؛ که این یک افزایش ناچیز در هزینه می‌باشد.

بنابراین بسته به مقادیر n و k ، اگر از دو جفت plaintext و ciphertext استفاده کنیم، احتمال بدست آوردن کلید کاذب کمتر شده و اطمینان بیشتری بدست می‌آید. همچنین در بدترین حالت نیاز است که 2^k کلید را چک کنیم.

۷.۳

ندانستن بردار اولیه (IV) به این معنی است که نمی‌دانیم چه برداری قبل از رمزگذاری با متن اصلی XOR شده است.

اگر دو جفت plaintext و ciphertext داشته باشیم؛ می‌توانیم از ciphertext بلوک اول به عنوان IV برای بلوک دوم استفاده کنیم و سپس مشابه با قسمت های قبلی، با جستجو کلید را بدست آورده و در نهایت اولین بلوک را توسط کلید رمزگشایی کرده و مقدار IV را بدست آوریم.

با داشتن جفت سوم plaintext و ciphertext، می‌توان نتایج را بررسی و سطح اطمینان بالاتری بدست آورده و همچنین احتمال بدست آوردن کلید کاذب را بسیار کمتر کنیم.

۷.۴

در حالتی که مقدار IV شناخته شده است، تنها تفاوت در هزینه محاسبه XOR برای هر بلوک در مد CBC است.

در حالتی که مقدار IV ناشناخته است، برای رسیدن به یک سطح اطمینان برابر، در مد CBC نیاز به یک جفت plaintext و ciphertext بیشتر نسبت به مد ECB داریم.

(به عبارتی دیگر با داشتن تعداد t جفت plaintext و ciphertext در هر دو مد، سطح اطمینان مد ECB برابر با t و سطح اطمینان مد CBC برابر با t-1 می‌باشد)

سوال ۸)

۸.۱

در حالت ECB، تنها بلوکی (Block) که تحت تاثیر قرار می‌گیرد؛ بلوک حاوی خطا می‌باشد، زیرا بلوک‌ها هیچ تعامل یا وابستگی به یکدیگر ندارند.

۸.۲

در حالت CBC، خطا در بلوک بعدی (y_{i+1}) نیز منتشر می‌شود، زیرا هنگام رمزگشایی بلوک y_{i+1}، تحت تاثیر خطای بلوک قبلی قرار می‌گیرد. ولی با توجه به این‌که بلوک y_{i+1} هنگام انتقال دچار خطا نشده‌است، بنابراین در هنگام رمزگشایی بلوک y_{i+2} خطا رخ نمی‌دهد.

۸.۳

در حالت CBC، یک خطا در متن اصلی (clear text) که قبل از عملیات رمزگذاری ایجاد شده‌است، تنها بر روی بلوک حاوی خطا تاثیر می‌گذارد؛ زیرا در این مورد، عملیات رمزگذاری و انتقال در واقع بدون خطا انجام شده و فقط داده‌های اشتباه رمزگذاری شده‌اند. بنابراین همان متن توسط باب دریافت می‌شود که توسط آلیس رمزگذاری شده ولی با معنایی نادرست. البته بلوک خطا و تمام بلوک‌های بعدی به دلیل انتشار این تغییر، متن رمز متفاوتی خواهند داشت، ولی این یک خطا نیست و این بلوک‌ها در سمت گیرنده به درستی رمزگشایی می‌شوند.

۸.۴

در حالت CFB، خطاها بر روی بلوکی که در آن رخ داده و بلوک بعدی، با خراب کردن کلید در هنگام رمزگشایی، تاثیر می‌گذارند. زیرا از متن رمزشده به عنوان ورودی در تولید کلید در عملیات رمزگذاری استفاده می‌شود و با توجه به این‌که CFB تا حدودی همانند یک رمز جریانی (stream cipher) عمل می‌کند؛ در بلوکی که خطا رخ می‌دهد، فقط آن بیت تغییر می‌کند. ولی با این وجود به دلیل فیدبک متن رمزشده حاوی خطا، بیت‌های کلید نیز دچار خطا شده و رمزگشایی بلوک بعدی (y_{i+1}) نیز حاوی خطا می‌باشد. ولی چون متن بلوک y_{i+1} خطا ندارد، بنابراین در هنگام رمزگشایی بلوک y_{i+2} خطا رخ نمی‌دهد (همانند حالت CBC).