به نام خدا

دانشکده مهندسی برق،
دانشگاه صنعتی شریف

مبانی رمزنگاری و امنیت شبکه

سیستم‌های رمز قالبی

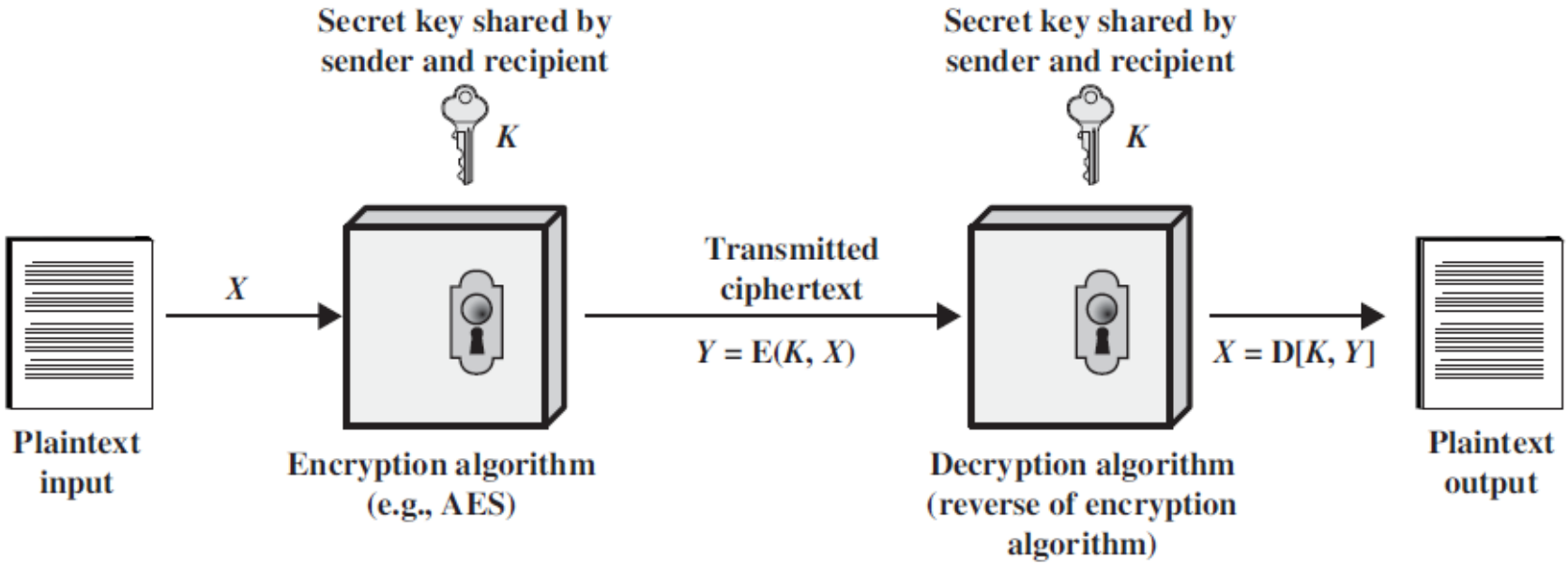
Block Ciphers

مهتاب میرمحسنی

نیم‌سال دوم (بهار) ۹۸-۹۹

سیستم رمز متقارن یا تک کلیدی (Symmetric=One Key)

- کلیدهای رمزگذاری و رمزگشایی یکسان یا به راحتی از روی یکدیگر قابل محاسبه
- سیستم‌های رمز قالبی (Block Ciphers)
- سیستم‌های رمز دنباله‌ای (Stream Ciphers)
- فرض: داده‌ها دنباله باینری {0,1}



سیستم‌های رمز دنباله‌ای (Stream Ciphers)

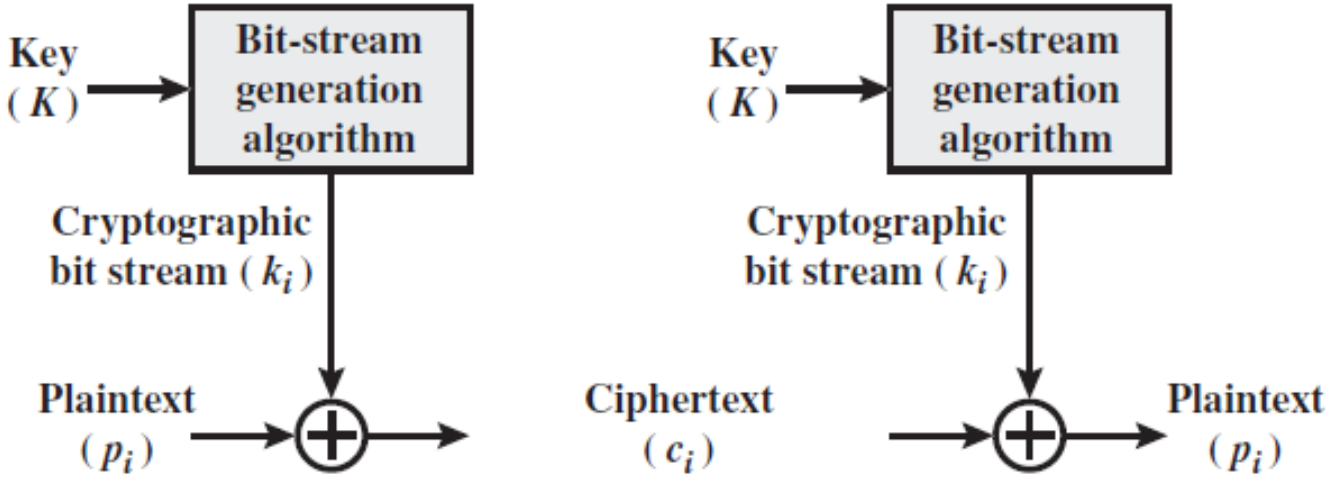
- هر سمبل از دنباله متن اصلی توسط سمبل متناظر دنباله کلید رمز می‌شود
  - معمولاً بیت یا بایت

$$P = P_1 P_2 P_3 \ldots$$

$$K = K_1 K_2 K_3 \ldots$$

$$C = E(K, P) = E(K_1, P_1) E(K_2, P_2) \ldots = C_1 C_2 \ldots \Rightarrow C_i = E(K_i, P_i)$$

- ایده‌آل: دنباله متن اصلی نامحدود ← دنباله کلید نامحدود
  - الگوریتم ساخت دنباله کلید نامتناهی از کلید اصلی

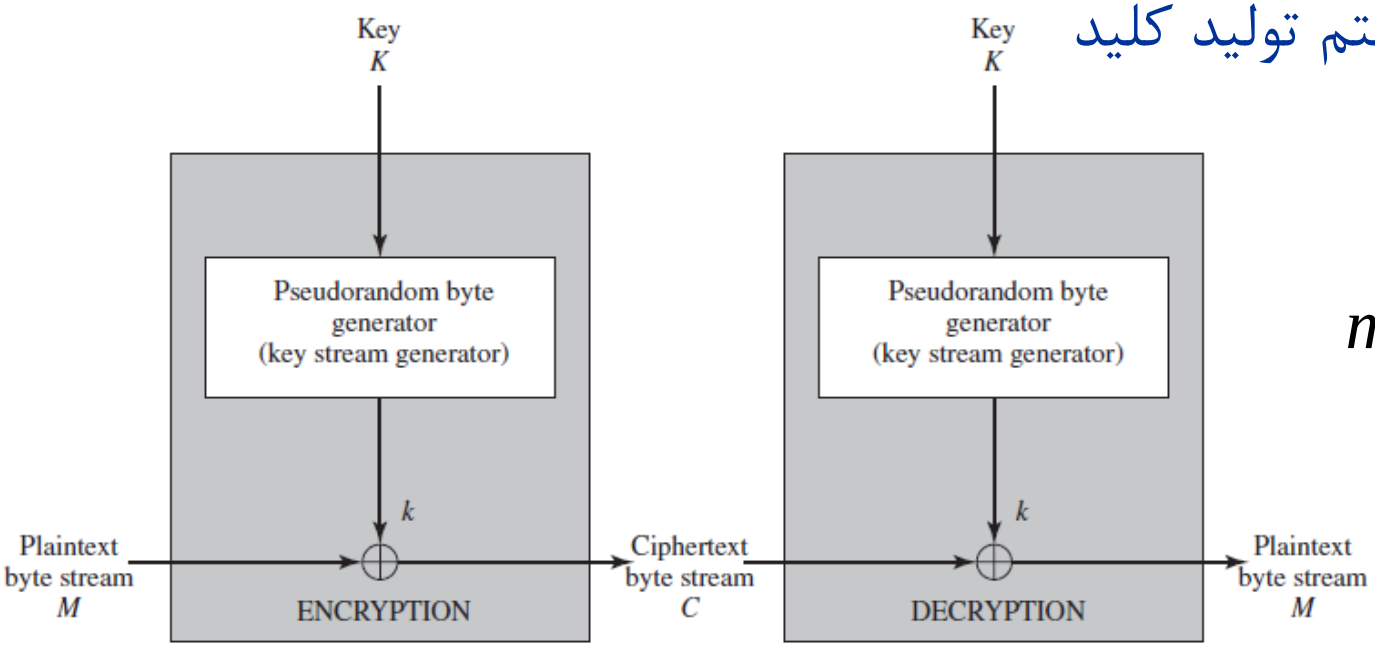


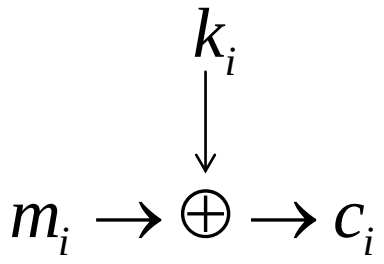
رمز دنباله‌ای

- دنباله کلید
  - نامتناوب: رمز ورنام ← امن کامل
  - متناوب: مانند رمز Vigenère

$$K = K_1 K_2 \ldots K_d K_1 K_2 \ldots K_d \ldots$$

- الگوریتم رمزگذاری ($E(K_i, \cdot)$) ساختار ساده XOR است (طبق معیار شانون)
- پیچیدگی: الگوریتم تولید کلید





رمز دنباله‌ای

- رمزگذاری $C_i = K_i \oplus P_i$

```
  11001100  plaintext
⊕ 01101100  key stream
  10100000  ciphertext
```

- رمزگشایی $P_i = K_i \oplus C_i$

```
  10100000  ciphertext
⊕ 01101100  key stream
  11001100  plaintext
```

$K_i = ?$

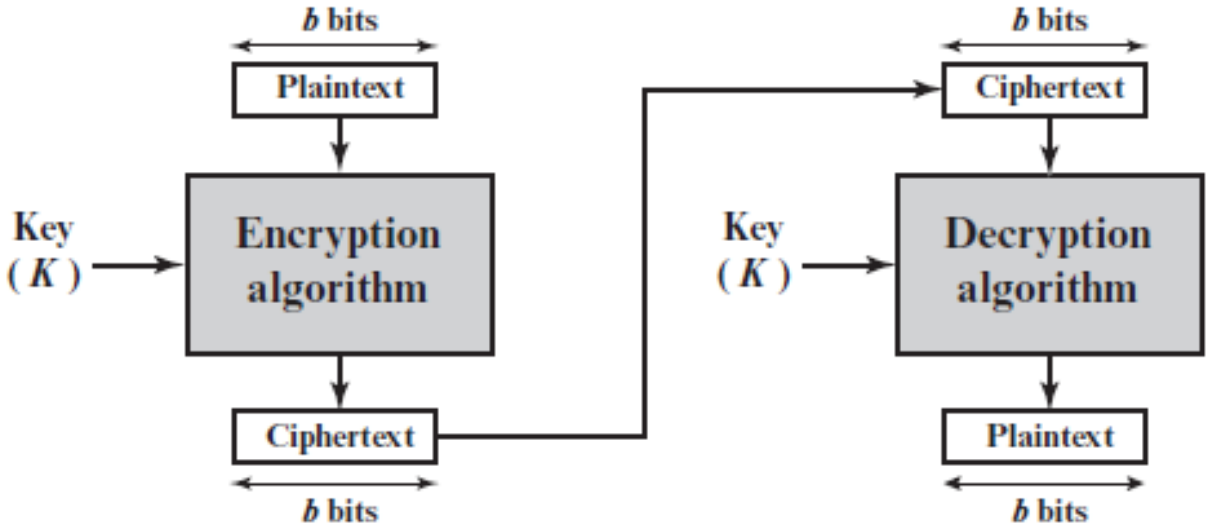
سیستم‌های رمز قالبی (Block Ciphers)

- در هر بار رمزگذاری روی قالبی به طول b توسط کلید رمزگذاری K انجام می‌شود

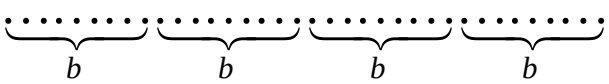
$$P = P_1 P_2 P_3 \ldots \qquad \overbrace{\cdots\cdots}^{b}\overbrace{\cdots\cdots}^{b}\overbrace{\cdots\cdots}^{b}\overbrace{\cdots\cdots}^{b}$$

$$C = E(K,P) = E(K,P_1)E(K,P_2)\ldots = C_1 C_2 \ldots \Rightarrow C_i = E(K,P_i)$$

- طول قالب معمول: ۶۴، ۱۲۸ یا ۲۵۶ بیت



سیستم‌های رمز قالبی (Block Ciphers)



$$C_i = E(K, P_i)$$

- سیستم رمز قالبی معادل یک سیستم رمز جانشینی ساده (تک الفبایی کلی) است
  - هر بار سمبلی به سمبل دیگر تبدیل می‌شود
  - شکسته نمی‌شود ← تعداد سمبل‌ها $2^b = 2^{128} \approx 10^{39}$
- پرکاربردتر از رمز دنباله‌ای
  - اکثر رمزهای متقارن بکار رفته در شبکه‌ها رمز قالبی هستند
- امنیت؟
  - با استفاده از مودهای کاری مشابه رمز دنباله‌ای

رمز دنباله‌ای و قالبی

- هر سیستم رمز کلاسیک یک سیستم رمز قالبی است
  - رمز Vigenère یک رمز قالبی به طول قالب d است
  - رمز Hill یک رمز قالبی به طول قالب m است
  - رمز جابجایی یک رمز قالبی به طول قالب d (جایگشت) است

- هر رمز دنباله‌ای متناوب یک رمز قالبی با طول قالب یک دوره تناوب است

$$E(K_1,\cdot)E(K_2,\cdot)\ldots E(K_d,\cdot) \triangleq E(K,\cdot)$$

- هر رمز قالبی یک رمز دنباله‌ای با دوره تناوب یک است
- کلاسیک: تعداد الفبا = خطی
- قالبی: تعداد الفبا = نمایی (2^b)، طول قالب = خطی
- دنباله‌ای: دوره تناوب = نمایی (معادل با طول قالب = نمایی)

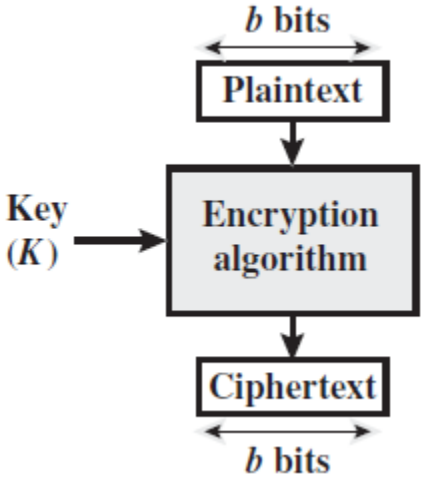
سیستم‌های رمز قالبی (Block Ciphers)

- مدرن
- یکی از پرکاربردترین الگوریتم‌های رمزنگاری
- خدمات امنیت و احراز اصالت

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)

سیستم‌های رمز قالبی (Block Ciphers)

- در هر بار رمزگذاری روی قالبی به طول n توسط کلید رمزگذاری K انجام می‌شود



$$\underbrace{\cdots\cdots}_{n}\underbrace{\cdots\cdots}_{n}\underbrace{\cdots\cdots}_{n}\underbrace{\cdots\cdots}_{n} \quad P = P_1P_2P_3\ldots$$

$$C = E(K,P) = E(K,P_1)E(K,P_2)\ldots = C_1C_2\ldots \Rightarrow C_i = E(K,P_i)$$

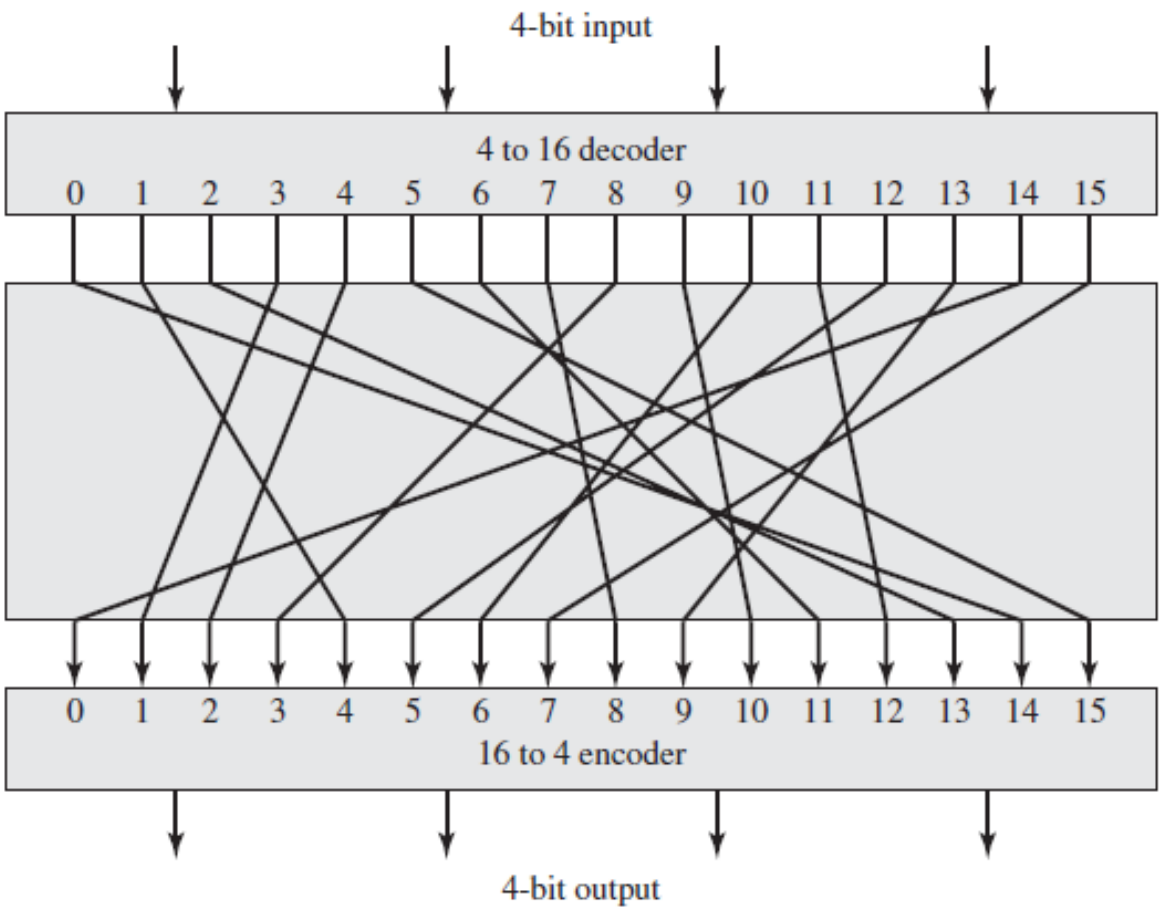
- طول قالب معمول: ۶۴، ۱۲۸ یا ۲۵۶ بیت
  - ۱۲۸ بیت ← اندازه حروف الفبا نسبت به طول متن اصلی نمایی است

$$2^{128} \approx 10^{39}$$

- عدم ابهام در آشکارسازی
- سیستم رمز قالبی یک سیستم رمز جانشینی ساده (تک الفبایی کلی) است
  - هر بار سمبلی به سمبل دیگر تبدیل می‌شود 2^n
  - فضای کلید (رمز تصادفی) $2^n!$

رمز تصادفی (جانشینی کلی)

- رمز قالبی **ایده‌آل**
- $n=4$



رمز تصادفی (جانشینی کلی)

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

- رمزگذاری و رمزگشایی
- طول قالب کوچک
  - سیستم جانشینی کلاسیک
  - آسیب‌پذیر در برابر حملات آماری
- طول قالب بزرگ
  - عملی نیست!
  - کلید = نگاشت = $n \times 2^n$ بیت
- n=64
  - مناسب برای مقابله با حملات آماری
  - طول کلید = $64 \times 2^{64} = 2^{70} \approx 10^{21}$ بیت که عملی نیست
- حل: زیرمجموعه‌ای از $2^n!$ نگاشت ممکن

رمز قالبی

- سیستم انتقال (سزار) $c_i = p_i + k$
  - از قالبی بودن ساختار استفاده چندانی نشد
- سیستم ضربی $c_i = A_{n\times n} p_i$
  - عدم ابهام در آشکارسازی: ماتریس ناویژه
- سیستم مستوی (affine) $c_i = A_{n\times n} p_i + t_{n\times 1}$

$$\|k\| = 2^n \times \left(2^n - 1\right) \times \left(2^n - 2\right) \times \left(2^n - 2^2\right) \times \cdots \times \left(2^n - 2^{n-1}\right)$$
$$\geq 2^{n(n-1)} \approx 2^{n^2}$$

- برخلاف سیستم‌های کلاسیک می‌توانند مفید واقع شوند
- سیستم خطی ← آسیب‌پذیری در برابر حمله نوع دوم ← استفاده از عوامل غیرخطی

روش پیشنهادی شانون

- اساس طراحی رمزهای قالبی مدرن
  - مقاله ۱۹۴۹ شانون
  - پراکنش (Diffusion): پراکنده کردن مشخصه آماری متن اصلی در متن رمز شده
  - آشفته‌سازی (Confusion): رابطه پیچیده میان کلید و متن رمز شده
- هدف: مقابله با تحلیل آماری توسط توزیع یکنواخت متن اصلی (فضای کوچک) روی کل فضای متن رمزشده

- رمز ترکیبی (Product Cipher): استفاده ترکیبی (تکرار) یک در میان از
  - جانشینی (substitution)← S-box
  - جایگشت (permutation)← P-box
- ساختار شبکه جانشینی–جایگشت (substitution- permutation network) SPN

تبدیلات مستوی

1. تکرار تبدیل مستوی، یک تبدیل مستوی است
2. تکرار تبدیل‌های مستوی همراه با جایگشت، یک تبدیل مستوی است
3. استفاده از تبدیل‌های Involution
   - معکوس آن‌ها خودشان است

$$\forall x,\ f\left(f\left(x\right)\right)=x$$

$$\forall x,\ f\left(x\right)=f^{-1}\left(x\right)$$

- از $m!$ تبدیل جابجایی متفاوت (طول قالب=m)، تقریبا $\sqrt{m!}$ از نوع involution است
- از $l!$ تبدیل جانشینی متفاوت (تعداد الفبا=l)، تقریبا $\sqrt{l!}$ از نوع involution است

- تکرار تبدیل‌های involution، یک تبدیل involution است

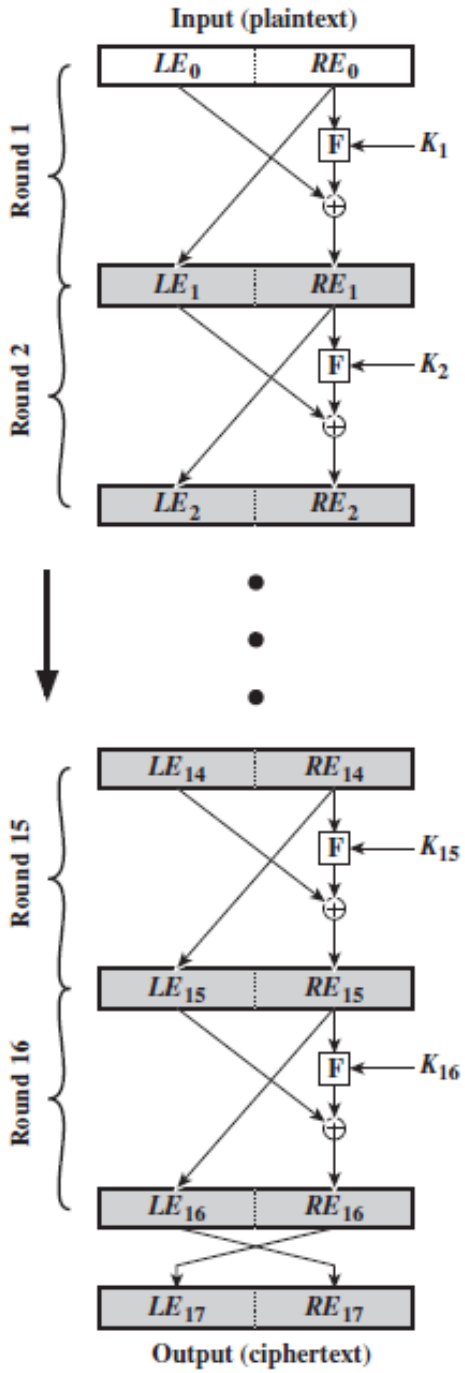
ملاحظات کلی در طراحی سیستم‌های رمز قالبی (SPN)
سیستم‌های فایستلی

1. از تبدیل‌های مخلوط (mixed) استفاده شود
   - جابجایی و جانشینی تواماً
2. تکرار در بکارگیری زیرتبدیل‌ها می‌تواند امنیت را افزایش دهد (حداقل دور)
   - نگاشت متن اصلی روی کل فضا
3. تبدیل‌های جابجایی و جانشینی یک در میان به کار رود
4. از عوامل غیرخطی در زیرتبدیل‌ها و یا تولید زیرکلیدها استفاده شود

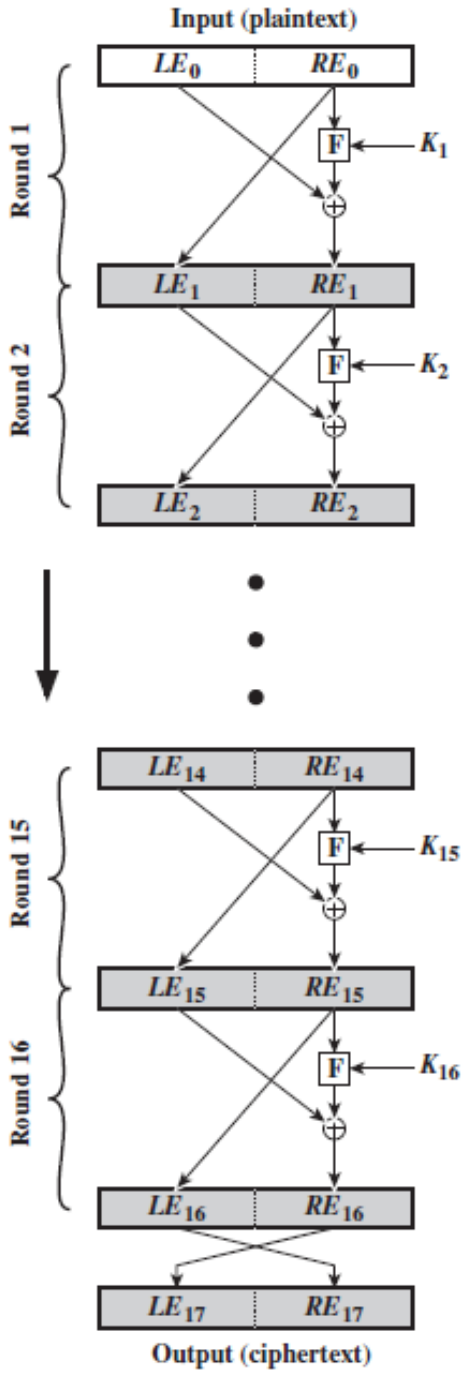
ملاحظات کلی در طراحی سیستم‌های رمز قالبی (SPN)
سیستم‌های فایستلی

5. در صورت امکان، از عوامل تصادفی یا شبه تصادفی در ساختار استفاده شود
6. تبدیل نهایی از نوع involution باشد
7. زیرالگوریتم‌ها تا جای ممکن ساده باشند و در خروجی آن‌ها کلید نقش اساسی داشته باشد
8. انعطاف‌پذیری طراحی الگوریتم
   - آزادی عمل به مصرف کننده
9. طول قالب از حداقلی بزرگ‌تر باشد
   - معیار Meyer (1970): حداقل طول قالب= ۴ حرف (۳۲ بیت)
   - ۱۶ حرف (۱۲۸ بیت) ← اندازه حروف الفبا نسبت به طول متن اصلی نمایی است

رمز فایستل (Feistel Cipher)
سیستم LUCIFER



- توسط Feistel در ۱۹۷۳
  - بر اساس پیشنهاد شانون در ۱۹۴۵
- استفاده از ایده‌های شانون، ترکیب، مخلوط و تکرار
  - حالت خاصی از SPN
- ورودی: $2w$ بیت متن اصلی و کلید K
  - طول قالب زوج و به دو قسمت تقسیم (L_0, R_0)
- پردازش در n دور
  - کلید $(k_1, k_2, \ldots, k_n)$
- زیرالگوریتم‌ها (دورها) یکسان
  - جانشینی و جایگشت



رمز فایستل (Feistel Cipher)

- اعمال جانشینی بر نیمه چپ
  - اعمال تابع دور F به نیمه راست و جمع (XOR) حاصل با نیمه چپ
- اعمال جایگشت
  - جابجایی دو نیمه داده
- دور *i*ام

$$\mu_{i-1} = (m_{i-1}, m_i) \rightarrow \mu_i = \left(m_i, m_{i-1} + F_{k_i}(m_i)\right) = (m_i, m_{i+1})$$

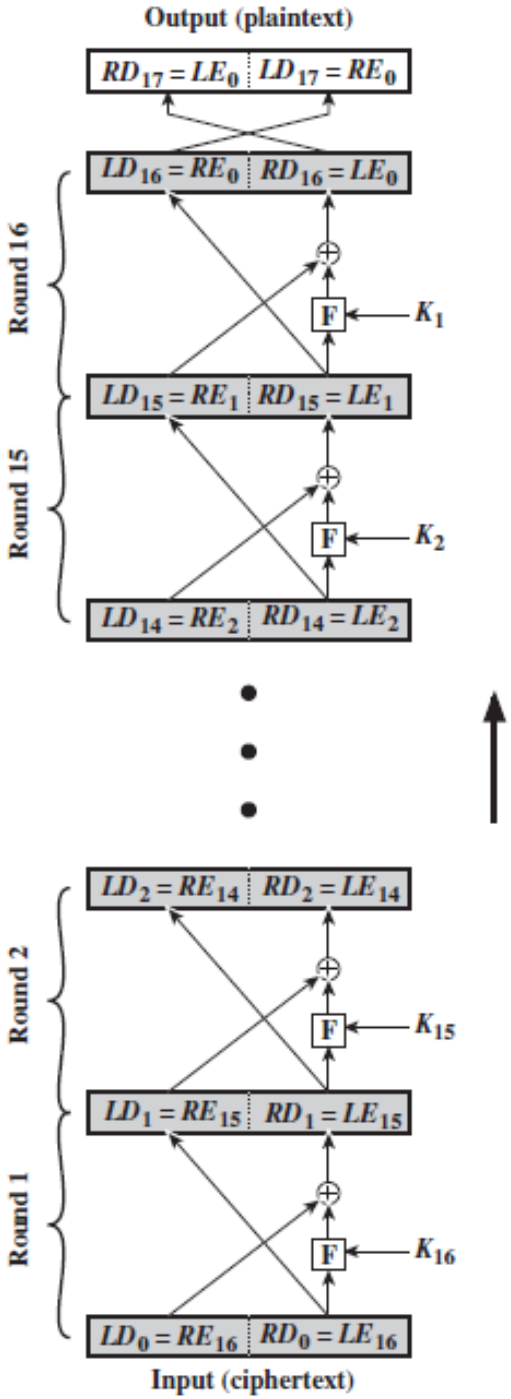
$$m_{i+1} = m_{i-1} + F_{k_i}(m_i)$$

$$m_{i-1} = m_{i+1} + F_{k_i}(m_i)$$

طراحی رمز فایستل

- طول قالب: ۶۴ یا ۱۲۸ بیت
  - بزرگ‌تر: امنیت بیشتر (diffusion) + سرعت رمزگذاری و رمزگشایی کمتر
- طول کلید: ۱۲۸ بیت یا بیشتر (۶۴ بیت دیگر کافی نسیت)
  - بزرگ‌تر: امنیت بیشتر (confusion و جستجوی فراگیر) + سرعت رمزگذاری و رمزگشایی کمتر
- تعداد دورها: معمولا ۱۶ دور
- الگوریتم تولید زیرکلید: هر چه پیچیده‌تر، مقابله با حملات رمزشکنی
- تابع دور F: هر چه پیچیده‌تر، تحلیل رمز سخت‌تر
- سرعت رمزگذاری و رمزگشایی
- سادگی تحلیل: هر چه ساده‌تر باشد، بررسی آسیب‌پذیری‌ها ساده‌تر
  - یکی از مشکلات DES: پیچیدگی تحلیل

رمزگشایی رمز فایستل



$$\mu_{i-1} = (m_{i-1}, m_i) \rightarrow \mu_i = \left(m_i, m_{i-1} + F_{k_i}(m_i)\right) = (m_i, m_{i+1})$$

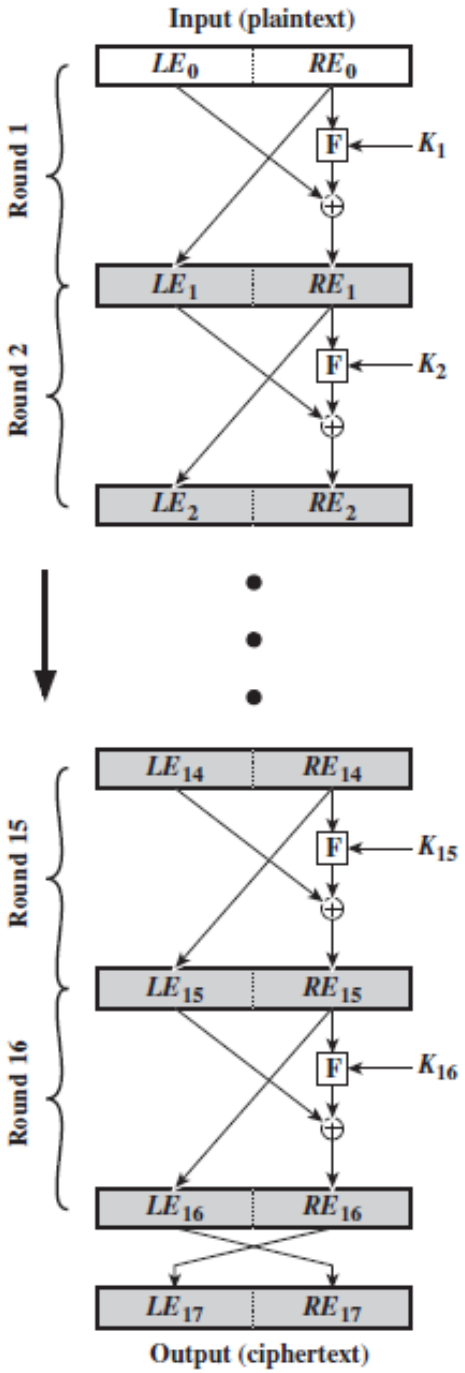
$$m_{i+1} = m_{i-1} + F_{k_i}(m_i) \quad \rightarrow \quad m_{i-1} = m_{i+1} + F_{k_i}(m_i)$$

- همان الگوریتم رمزگذاری ولی ترتیب زیرکلیدها عکس

$$\mu_i = (m_{i+1}, m_i) \rightarrow \mu_{i-1} = \left(m_i, m_{i+1} + F_{k_i}(m_i)\right) = (m_i, m_{i-1})$$

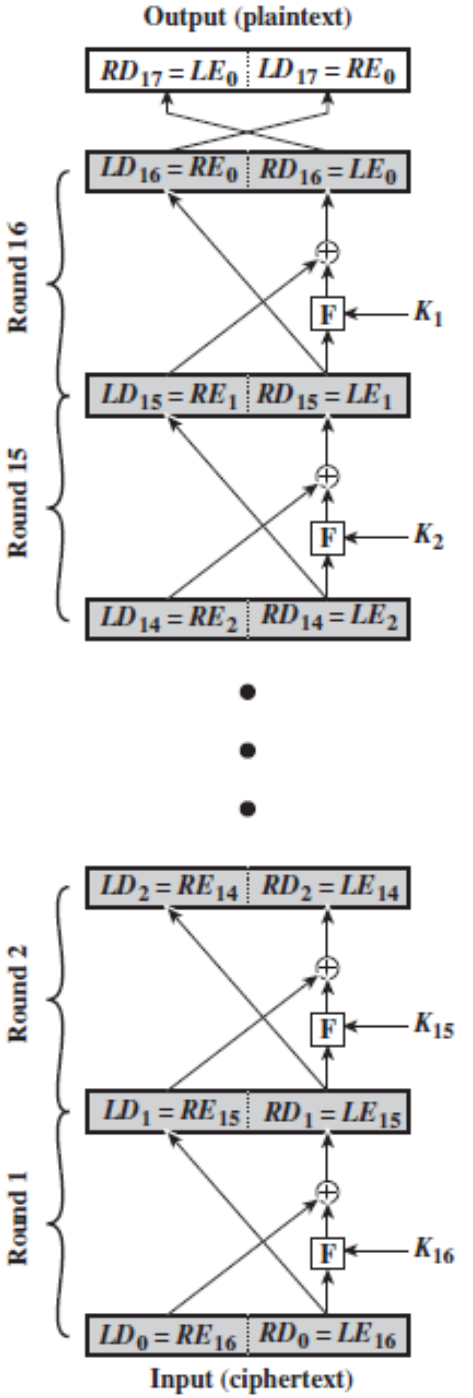
- تبدیل فایستل از نوع involution است
  - تکرار تبدیل involution
  - لزومی ندارد F یک به یک باشد

رمز فایستل



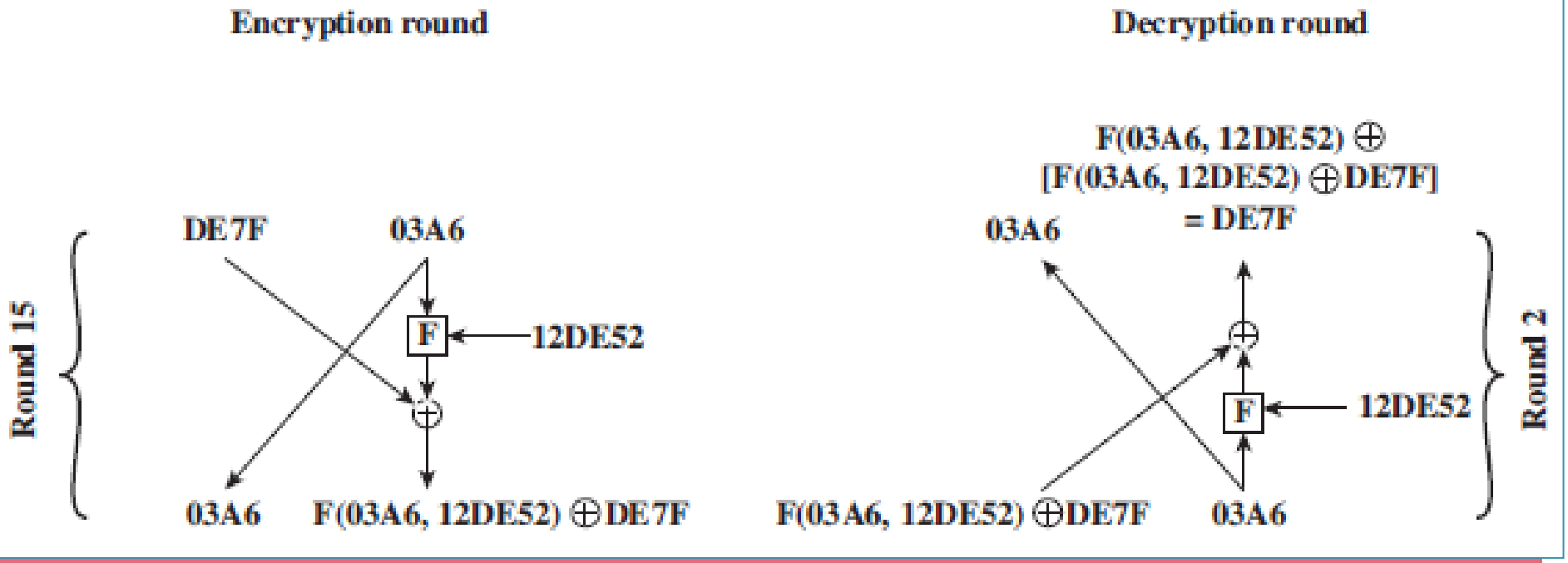
$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus \mathrm{F}(RE_{i-1}, K_i)$$

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus \mathrm{F}(RE_{i-1}, K_i)$$
$$= RE_i \oplus \mathrm{F}(LE_i, K_i)$$



مثال رمز فایستل

- دور ۱۵ام رمزگذاری (معادل دور ۲ رمزگشایی)
- طول قالب = ۳۲ بیت
- طول کلید = ۲۴ بیت

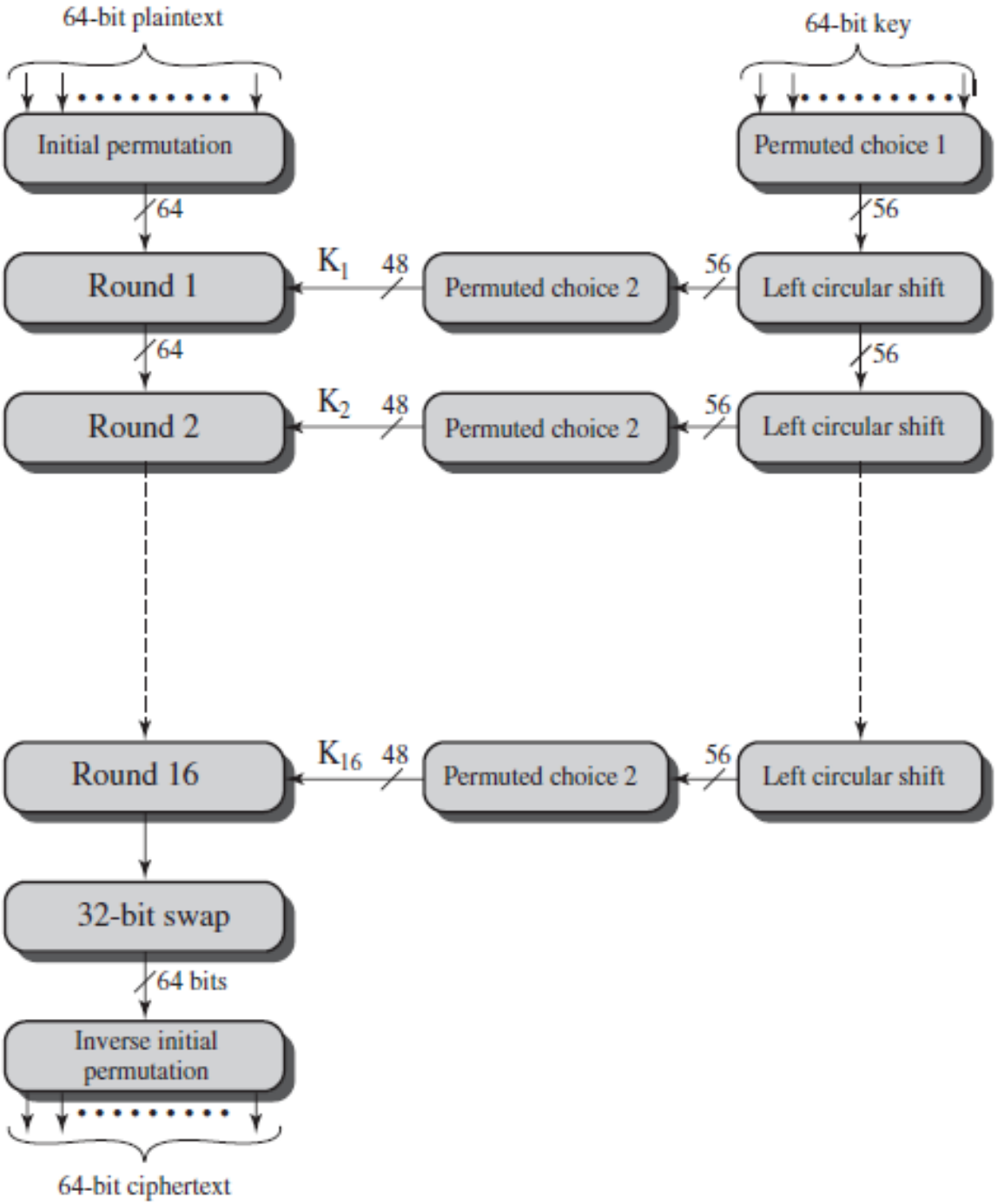


Data Encryption Standard (DES)

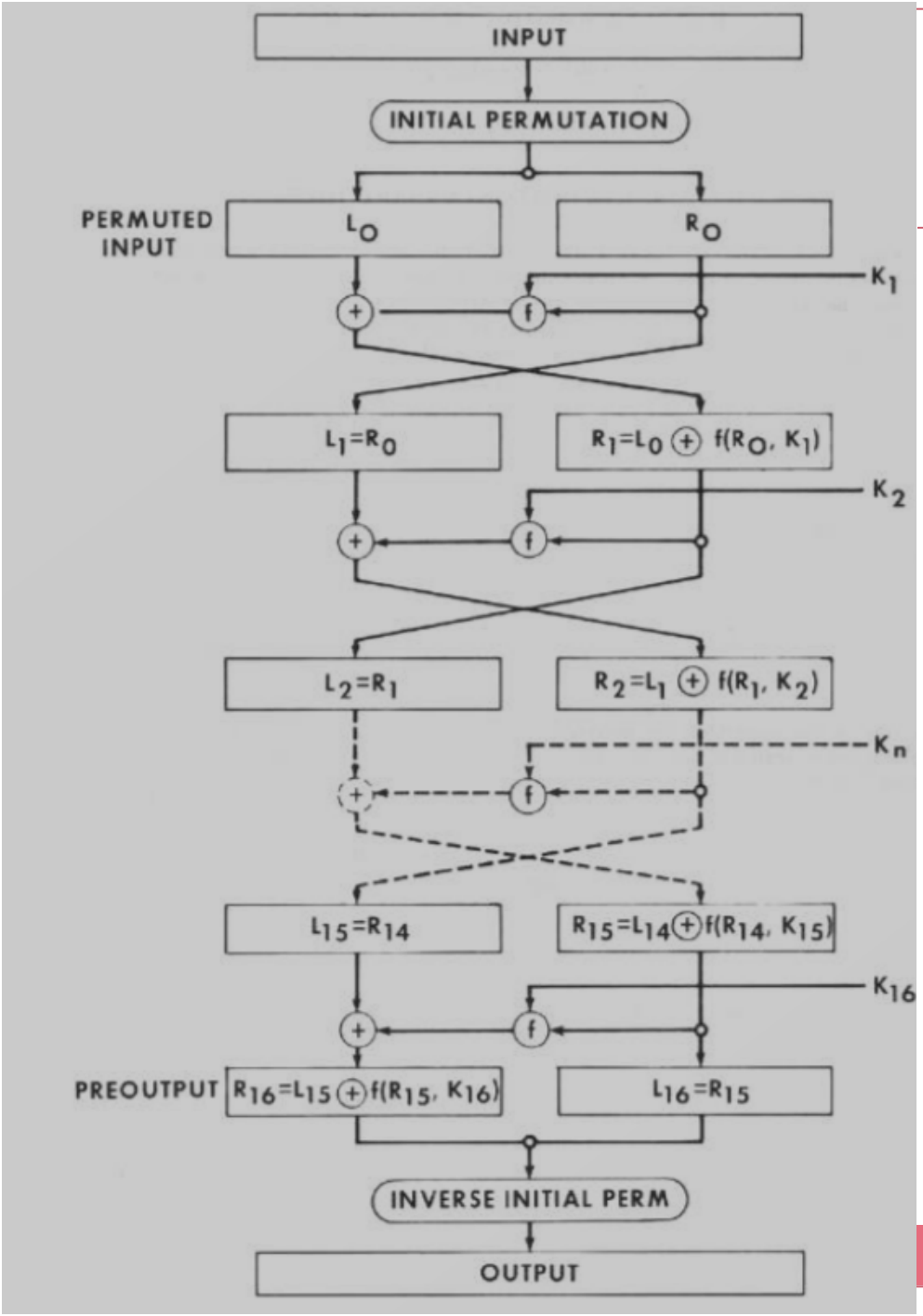
- بر پایه سیستم lucifer
  - طول قالب = ۶۴ بیت، طول کلید = ۱۲۸ بیت
- نمونه تجاری
  - Walter Tuchman and Carl Meyer (۱۹۷۷)
    - استاندارد FIPS PUB 46 توسط NBS (NIST)
  - طول قالب = ۶۴ بیت، طول کلید = ۵۶ بیت، تعداد دور = ۱۶
- کاربرد گسترده
- الگوریتم‌ها معلوم هستند ولی اصول طراحی و مبانی نظری آن‌ها فاش نشد!
- بحث درباره امنیت
  - طول کلید در برابر حمله جستجوی فراگیر کوچک است
  - مخفی بودن اصول طراحی (به ویژه s-boxها)
    - بررسی‌های آتی نشان داده که طراحی مناسب است

ساختار DES

- ورودی
  - متن اصلی (۶۴ بیت)
  - کلید ۶۴ بیت
    - ۵۶ بیت کلید
    - ۸ بیت پریتی (و یا دلخواه)



ساختار DES



$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

جایگشت اولیه (initial permutation)

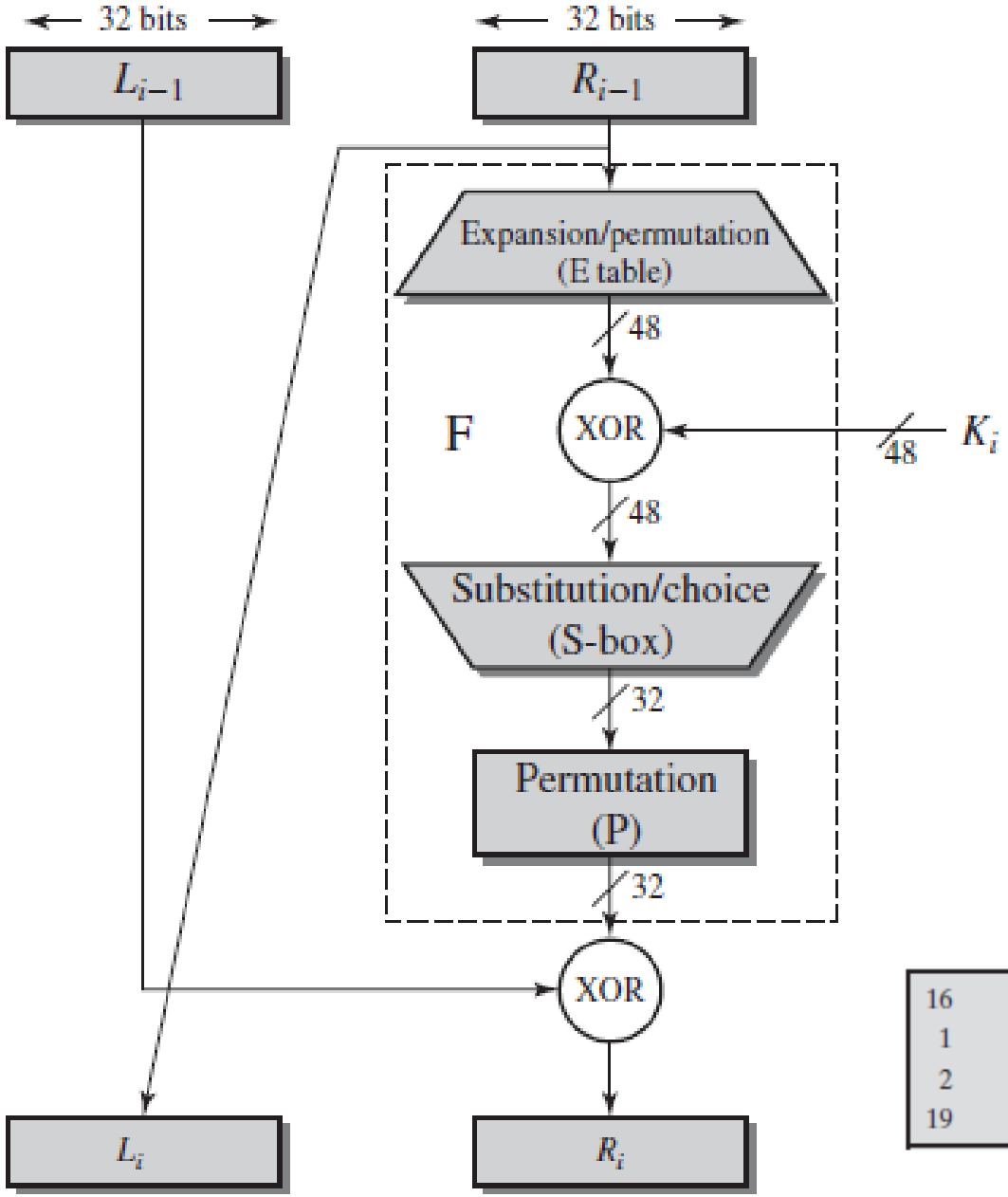
(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (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

یک دور DES



$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

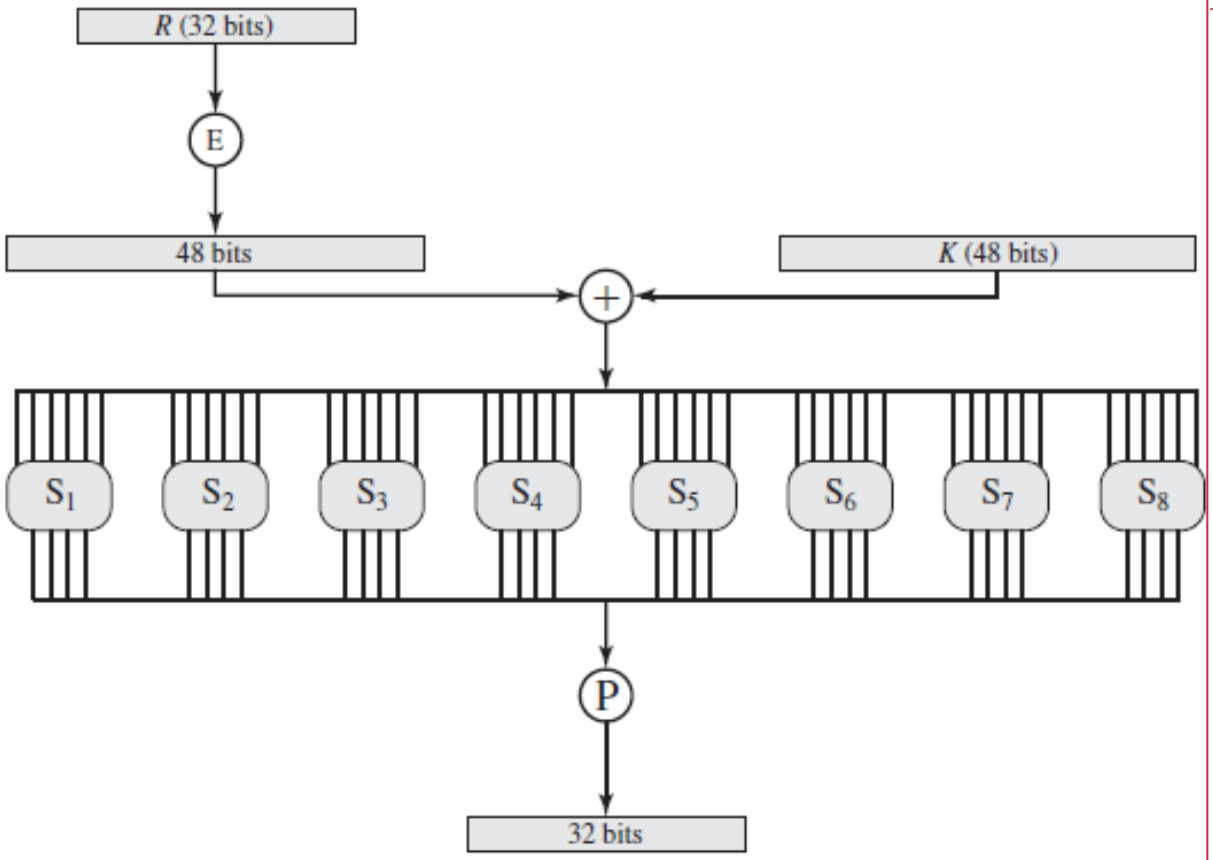
(c) Expansion Permutation (E)

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

(d) Permutation Function (P)

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

تابع دور



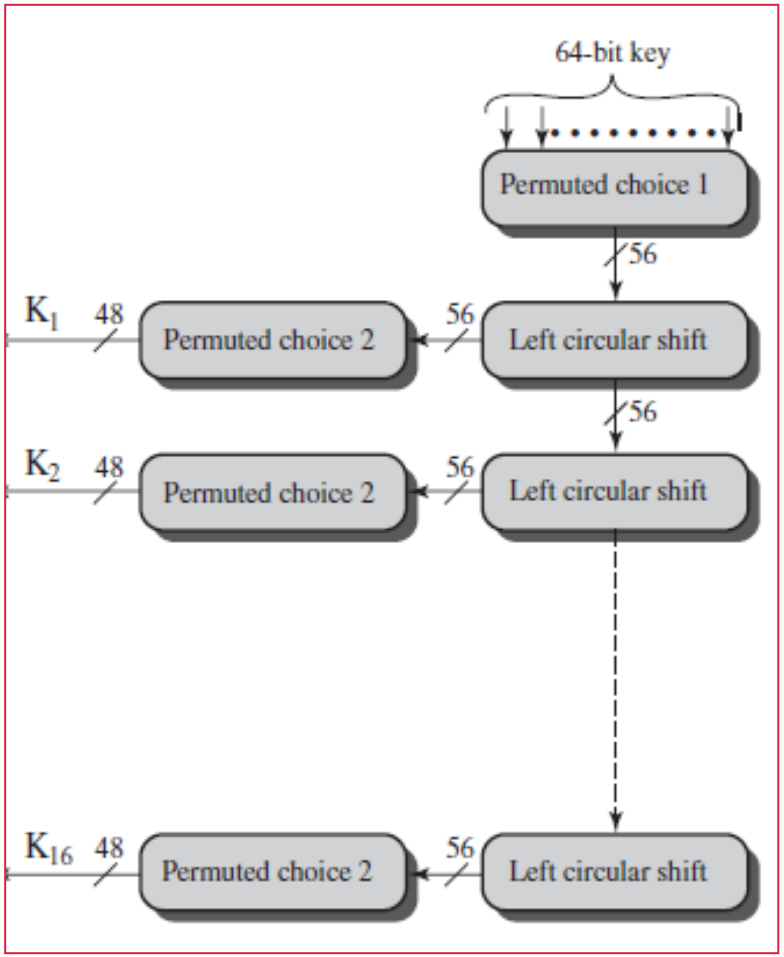
- ۸ تا s-box
- ۶ بیت ورودی و ۴ بیت خروجی (معکوس ناپذیر)
- بخش غیرخطی الگوریتم
- دو بیت اول و آخر ورودی
  - انتخاب یکی از ۴ سطر (۰-۳) جدول (الفبای جانشینی)
- ۴ بیت میانی
  - انتخاب یکی از ۱۶ ستون (۰ تا ۱۵)

$\overline{0}11001\overline{0}$

1001

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

الگوریتم تولید زیرکلید



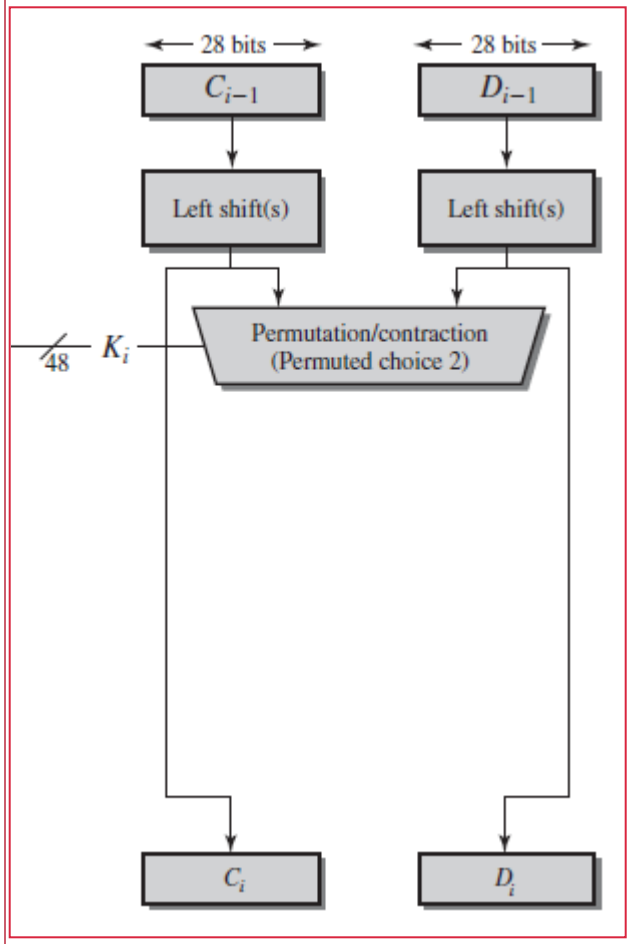
(a) Input Key

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

(b) Permuted Choice One (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

الگوریتم تولید زیرکلید



- شیفت چرخشی به چپ (۱ یا ۲ بیت)

(d) Schedule of Left Shifts

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

(c) Permuted Choice Two (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

- حذف ۹، ۱۸، ۲۲، ۲۵، ۳۵، ۳۸، ۴۳، ۵۴

رمزگشایی DES

- ساختار فایستلی (Involution)
- همان الگوریتم رمزگذاری ولی ترتیب زیرکلیدها عکس می‌شود
- جایگشت اولیه و نهایی نیز جابجا می‌شوند

تابع دور

- هر سطر s-box یک تبدیل جانشینی کلی معکوس‌پذیر

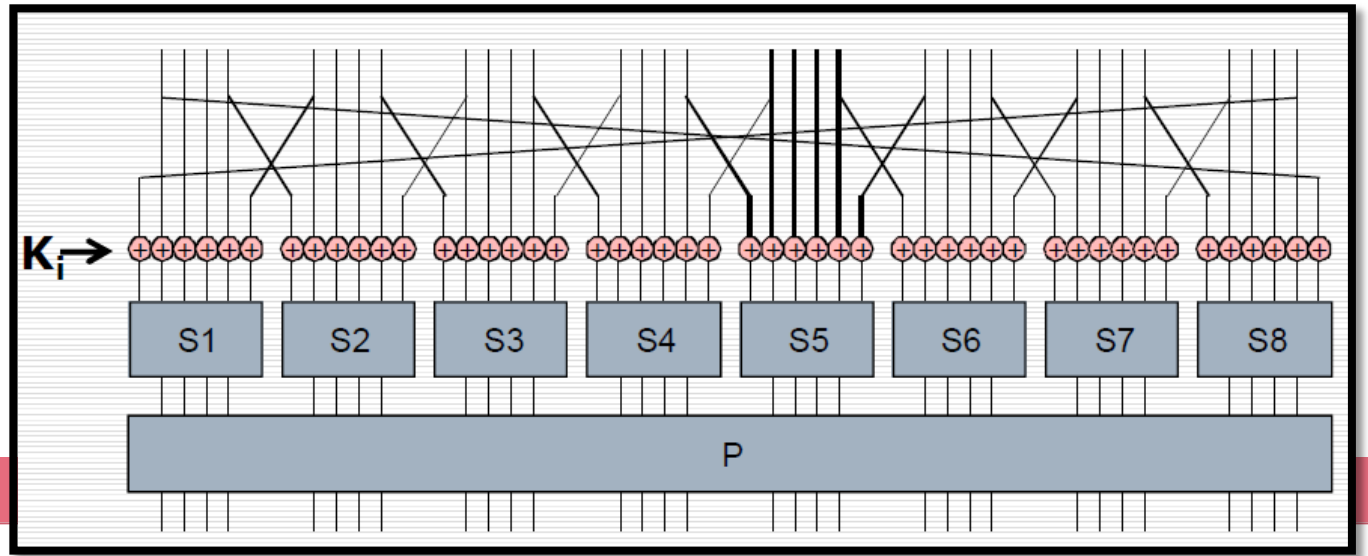
S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- بررسی جدول expansion: دو بیت کناری در ورودی s-box از ۴ بیت‌های کناری انتخاب می‌شوند

```
... efgh ijkl mnop ...  ⟶  ... defghi hijklm lmnopq ...
```

- اعمال جایگشت در خروجی: در دور بعدی روی تعداد بیشتری از بیت‌ها تاثیر می‌گذارد



اصول طراحی s-boxها

- هیچ‌کدام از s-boxها قالبی خطی یا مستوی نیستند (تنها قسمت غیرخطی)
- تغییر یک بیت ورودی، حداقل ۲ بیت خروجی را تغییر می‌دهد
- $S(x)$ و $S(x+001100)$ حداقل در ۲ بیت متفاوتند
  - تغییر دو بیت میانی ورودی، حداقل ۲ بیت خروجی را تغییر می‌دهد
- برای بیت‌های اختیاری e_1 و e_2 داریم: $S(x) \neq S(x \oplus 11e_1e_200)$
- هر سطر تمامی ۱۶ خروجی ممکن را داشته باشد
- مقابله با حملات تحلیل رمز تفاضلی و افزایش confusion

اصول طراحی جایگشت (P)

- ۴ بیت خروجی هر s-box در دور iام
  - ۲ بیت، بیت‌های میانی دور $i+1$ام را تشکیل می‌دهند ← با s-box های دیگر مشترک نیستند
  - ۲ بیت، بیت‌های کناری دور $i+1$ام را تشکیل می‌دهند ← با s-box های کناری مشترک هستند
- ۴ بیت خروجی هر s-box در دور iام، بر ۶ s-box متفاوت در دور $i+1$ام تاثیر می‌گذارد و هیچ دوتایی بر s-box یکسانی تاثیر ندارند
- اگر یک بیت خروجی S_j بیت میانی S_k را تحت تاثیر قرار دهد، در دور بعد بیت خروجی S_k نباید بیت میانی S_j را تحت تاثیر قرار دهد
  - بیت خروجی S_j نباید بیت میانی S_j را تحت تاثیر قرار دهد

اثر بهمنی (The Avalanche Effect)

- تغییر یک بیت ورودی (یا کلید)، تعداد زیادی از بیت‌های خروجی را تغییر دهد

Table 3.6 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

اثر بهمنی (The Avalanche Effect)

- تغییر یک بیت ورودی (یا کلید)، تعداد زیادی از بیت‌های خروجی را تغییر دهد

Table 3.6 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e	18
4		
5		
6		
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33
9	c11bfc09887fbc6c	32

0f1571c947d9e859

1f1571c947d9e859

Table 3.7 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

حملات DES

- **حمله جستجوی فراگیر**
- طول کلید = ۵۶ بیت $\|K\| = 2^{56} \approx 7.2 \times 10^{16}$
- ۱ تراشه: هر رمزنگاری DES در ۱ میکروثانیه⟵ بیش از ۱۰۰۰ سال
- ۱ میلیون تراشه، ۱ میکروثانیه⟵ ۱۰ ساعت
  - هزینه (در ۱۹۷۷): ۲۰ میلیون دلار
  - ۱۹۹۸: ۲۵۰ هزار دلار ⟵ توسط EFF در کمتر از ۳ روز شکسته شد
  - ۱۹۹۹: با همکاری اینترنتی در ۲۳ ساعت (توسط ۱۰۰۰ کامپیوتر) شکسته شد
- اطلاعات اولیه در مورد متن اصلی (قابل فهم)
  - روش نرم‌افزاری درک متن اصلی

حمله جستجوی فراگیر

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

حملات DES

- نگرانی از حملات تحلیل رمز به s-boxها
  - اصول (تئوری) طراحی نامشخص است
  - تا کنون مورد مهمی گزارش نشده!

- **حملات زمانی** (Timing Attacks)
  - بیشتر در مورد رمزهای کلید همگانی
  - پیاده‌سازی را مورد هدف قرار می‌دهد
  - با مشاهده زمان رمزگشایی یک متن رمز شده، اطلاعاتی در مورد کلید و متن اصلی بدست می‌آید
  - به نظر می‌رسد، DES در برابر حمله زمانی نسبتا مقاوم است (ولی نیاز به بررسی بیشتر دارد)

حملات تحلیلی DES

- حمله تحلیلی به ساختار داخلی DES
  - با توجه به افزایش طول کلید در استانداردهای بعدی (3-DES و AES) از محبوبیت بیشتری برخوردار شده‌اند
  - با یافتن اطلاعاتی در مورد رمزنگاری، برخی از بیت‌های زیرکلیدها را می‌یابند
  - در ادامه با حمله جستجوی فراگیر سایر بیت‌ها مشخص می‌شوند
- تحلیل رمز تفاضلی (differential cryptanalysis)
- تحلیل رمز خطی (linear cryptanalysis)
- ترکیب حمله تفاضلی و خطی

تحلیل رمز تفاضلی (Differential Cryptanalysis)

- یکی از قوی‌ترین حملات به سیستم‌های قالبی در حال حاضر
  - حمله آماری به ساختار فایستلی
  - Biham & Shamir 1990-93 – Murphy 1990
- حمله نوع سوم: حمله متن اصلی منتخب (Chosen Plaintext Attack)
  - نیاز به 2^{47} متن اصلی منتخب دارد
- طراحان DES از این حمله مطلع بوده‌اند (از ۱۹۷۴)
  - طراحی s-box و جایگشت
  - حمله تفاضلی به سیستم LUCIFER با ۸ دور نیاز به ۲۵۶ متن اصلی منتخب دارد
  - حمله تفاضلی به سیستم DES با ۸ دور نیاز به 2^{14} متن اصلی منتخب دارد
- تفاضل ورودی با چه احتمالی تبدیل به تفاضل مشخصی در خروجی می‌شود

$$\Delta m = m \oplus m' \xrightarrow{\text{Prob } p} \mathrm{E}(K, m) \oplus \mathrm{E}(K, m')$$

تحلیل رمز خطی (Linear Cryptanalysis)

- Matsui 1993
- یافتن تقریب خطی برای الگوریتم DES
- نیاز به 2^{43} متن اصلی معلوم (حمله نوع دوم) دارد
  - باز هم عملی نیست!
- جستجو برای معادلات خطی میان متن اصلی، متن رمزشده و کلید
  - احتمال p

$$A[i, j, \ldots, k] = A[i] \oplus A[j] \oplus \ldots \oplus A[k]$$

$$P[\alpha_1, \alpha_2, \ldots, \alpha_a] \oplus C[\beta_1, \beta_2, \ldots, \beta_b] = K[\gamma_1, \gamma_2, \ldots, \gamma_c]$$

اصول طراحی رمزهای قالبی (فایستلی)

- **تعداد دور**
  - هرچه بیشتر، تحلیل رمز مشکل‌تر (حتی برای تابع دور ضعیف)
  - معیار: حملات تحلیل رمز شناخته شده پیچیده‌تر از حمله جستجوی فراگیر باشند
  - DES با ۱۶ دور: حمله تفاضلی $2^{55.1}$ عملیات و حمله جستجوی فراگیر 2^{55} عملیات
- **تابع دور (F)**
  - هسته رمز فایستلی
  - هر چه غیرخطی‌تر، تحلیل رمز مشکل‌تر (تقریب آن با معادلات خطی مشکل باشد)
  - اثر بهمنی (strict avalanche criterion (SAC))
  - استقلال بیت‌های خروجی (bit independence criterion (BIC))
- **الگوریتم تولید زیرکلید**
  - کمتر بررسی شده

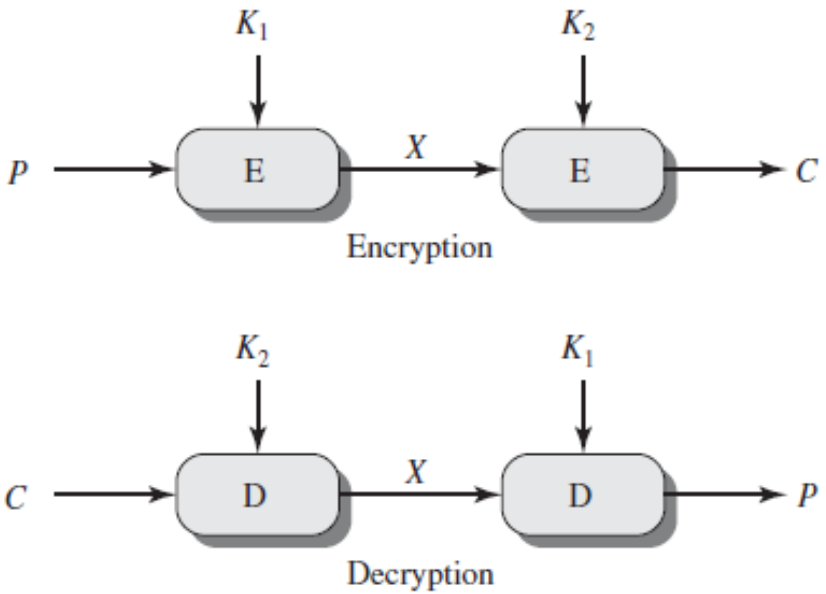
امن کردن DES

- طول کلید ← آسیب‌پذیر در مقابل حمله جستجوی فراگیر
  - طراحی الگوریتم جدید (AES)
  - تکرار الگوریتم (triple DES)

- **Double DES**

$$\|K\| = 56 \times 2 = 112 \text{ bits}$$

$$C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$$
$$P = \mathrm{D}(K_1, \mathrm{D}(K_2, C))$$

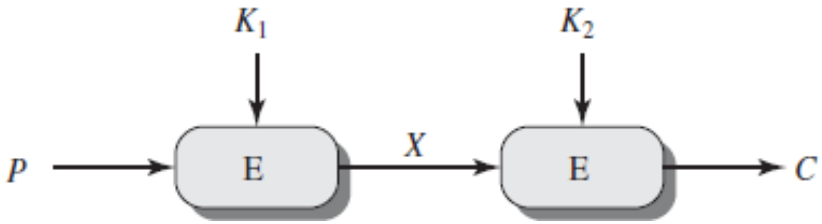


Double DES (2-DES)

- اگر: $$\mathrm{E}(K_2, \mathrm{E}(K_1, P)) = \mathrm{E}(K_3, P)$$
  - تکرار DES منجر به یک الگوریتم با ۵۶ بیت کلید می‌شد و در نتیجه بی‌فایده بود
- قالب: ۶۴ بیتی
  - تصادفی $(2^{64})! = 10^{34738000000000000000} > (10^{10^{20}})$
  - DES $2^{56} < 10^{17}$
- دو بار اعمال DES منجر به یکی از نگاشت‌ها (به غیر از DES) می‌شود
  - در سال ۱۹۹۲ اثبات شد
- **حمله ملاقات در میانه (Meet-in-the-middle)**
- ربطی به ساختار DES ندارد و قابل اعمال به تمام رمزهای قالبی تکراری است
  - دیفی و هلمن ۱۹۷۷

حمله ملاقات در میانه (Meet-in-the-middle)

$$X = \mathrm{E}(K_1, P) = \mathrm{D}(K_2, C)$$

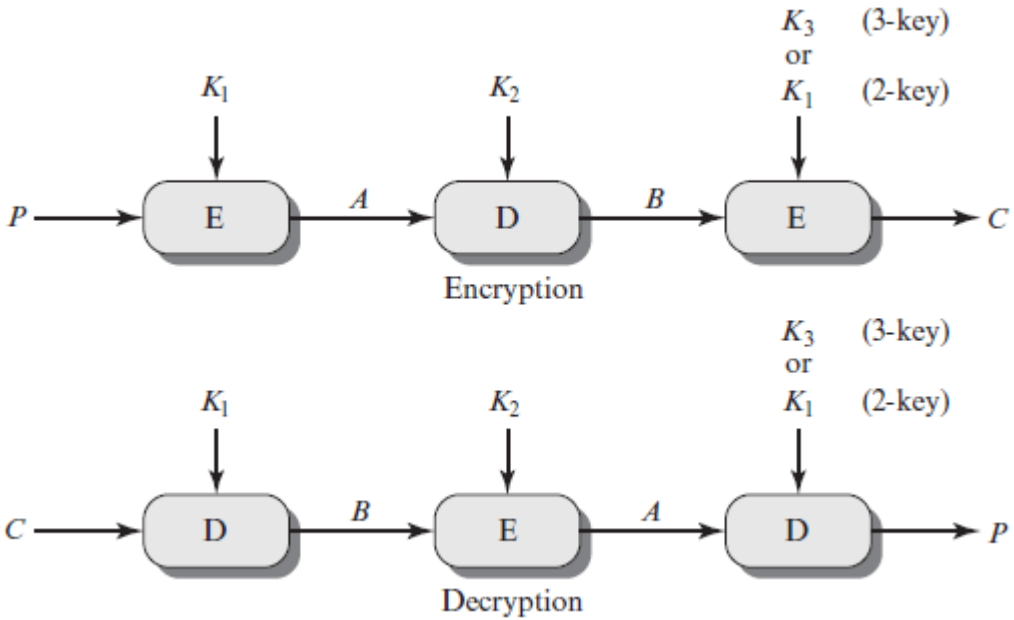


- در حمله نوع دوم با دانستن (P,C)
- P را با تمامی 2^{56} کلید ممکن (K_1) رمزگذاری و نتایج را (برای X) ذخیره کن
- C را با تمامی 2^{56} کلید ممکن (K_2) رمزگشایی و نتایج را با X ذخیره شده مقایسه کن
- در صورت یافتن تطابق (در X)، مقادیر کلیدهای پیدا شده را با استفاده از زوج جدید (P,C) امتحان کن
- پیچیدگی حمله از مرتبه 2^{56} است.
- اگر روی دو زوج (P,C) صدق کند: احتمال false alarm $= 2^{-16}$
- مقابله: استفاده از ۳ تکرار با ۱۶۸ بیت کلید و امنیت از مرتبه 2^{112}
  - کلید طولانی: راه حل استفاده از ۳ تکرار با ۲ کلید است (Tuchman, 1979)
  - هر دو استاندارد ۲ و ۳ کلید وجود دارد

Triple DES (3-DES)
با دو یا سه کلید

$$C = \mathrm{E}(K_1, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$

$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_1, C)))$$



- مزیت استفاده از رمزگشایی (D): پیاده‌سازی DES با استفاده از 3-DES

$$C = \mathrm{E}(K_1, \mathrm{D}(K_1, \mathrm{E}(K_1, P))) = \mathrm{E}(K_1, P)$$

$$P = \mathrm{D}(K_1, \mathrm{E}(K_1, \mathrm{D}(K_1, C))) = \mathrm{D}(K_1, C)$$

Triple DES (3-DES)
با دو کلید

- استانداردهای مدیریت کلید
  - ANS X9.17 و ISO 8732
- حمله جستجوی فراگیر
  - از مرتبه 2^{112}
- حمله تفاضلی
  - از مرتبه 10^{52}
- حمله نوع سوم (حمله متن اصلی منتخب) + حمله ملاقات در میانه
  - از مرتبه 2^{56} ولی نیاز به 2^{56} متن اصلی منتخب دارد
- حمله نوع دوم (حمله متن اصلی معلوم): کاهش به 2-DES
  - از مرتبه $2^{120-\log_2 n}$

Triple DES (3-DES)
با سه کلید

- هرچند حملات به 3-DES با دو کلید عملی نیست، استفاده از ۳ کلید امروزه ترجیح دارد
- طول کلید=۱۶۸ بیت ولی معادل ۱۱۲ بیت امنیت دارد

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$

- مدیریت کلید SP 800-57
- استفاده در کاربردهای اینترنتی
- استانداردهای امنیت پست الکترونیکی
  - PGP
  - S/MIME

Advanced Encryption Standard (AES)

- در سال ۲۰۰۱ توسط National Institute of Standards and Technology (NIST) انتشار یافت
- در بسیاری از استانداردها جایگزین DES شد
- مسابقه‌ای توسط NIST (با هدف ارتقای امنیت DES) در سال ۱۹۹۷ برگزار شد و فینالیست‌ها (۵ تا از ۱۵ الگوریتم ارسالی):

1. Rijndael: 86 positive, 10 negative → Rijmen & Daemen
2. Serpent: 59 positive, 7 negative → Anderson, Biham, Knudsen
3. Twofish: 31 positive, 21 negative → Schneier, Kelsey, Whiting, Wagner, Hall, Ferguson
4. RC6: 23 positive, 37 negative → RSA
5. MARS: 13 positive, 84 negative → IBM

ویژگی‌های AES

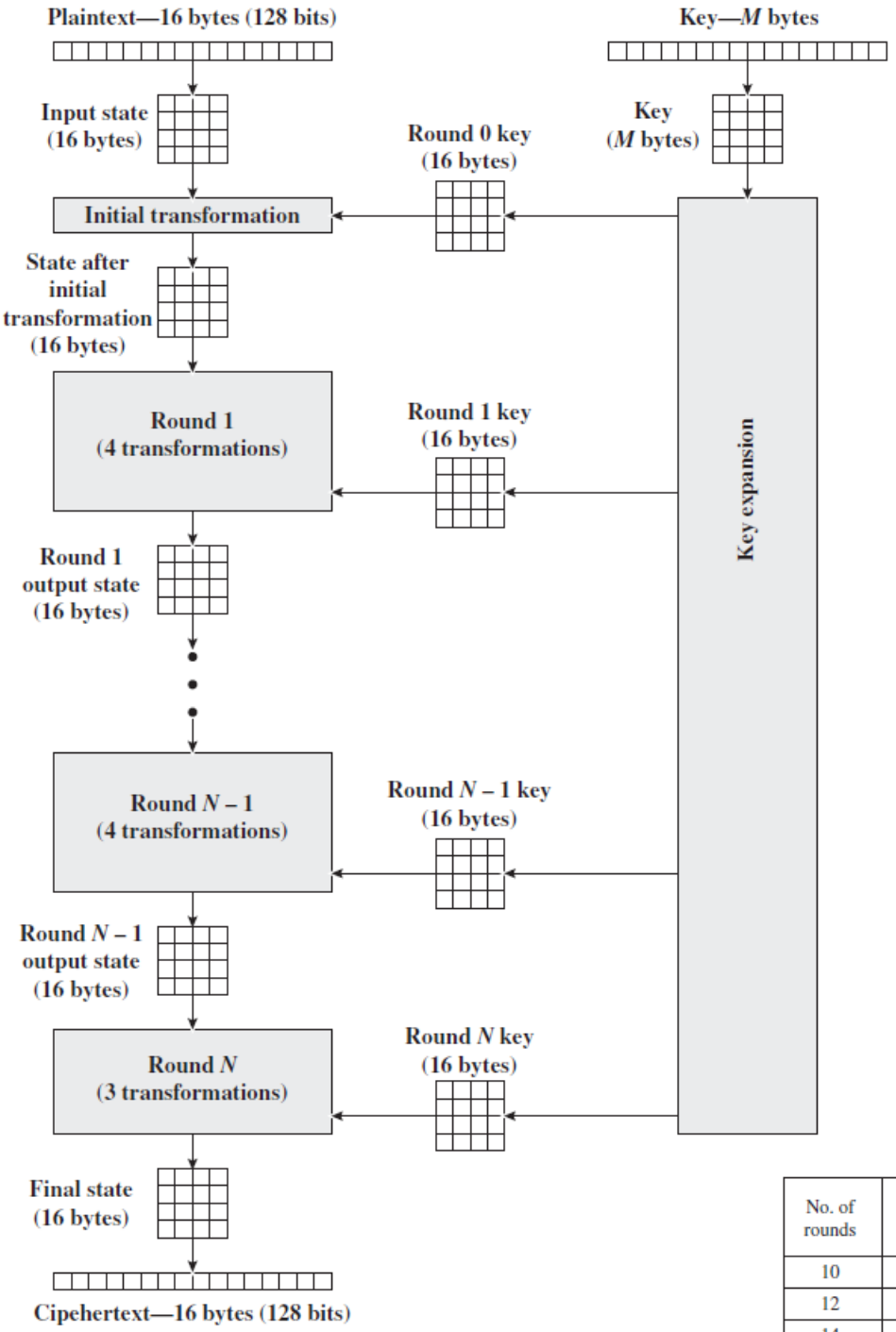
- سادگی (تحلیل)
- انعطاف‌پذیری (الگوریتم و کلید)
- پیاده‌سازی (نرم‌افزاری و سخت‌افزاری)
- امنیت (مقابله با حملات)

مشخصات AES

- عملیات بایتی در میدان $GF(2^8)$
- چندجمله‌ای ساده نشدنی $m(x) = x^8 + x^4 + x^3 + x + 1$
- طول قالب = ۱۲۸ بیت (۱۶ بایت) در استاندارد FIPS PUB 197
  - پیشنهاد Rijndael طول قالب‌های ۱۲۸، ۱۹۲ و ۲۵۶ بیت بود
- طول کلید = ۱۲۸، ۱۹۲ و ۲۵۶ بیت (۱۶، ۲۴ و ۳۲ بایت)⟵ (۴، ۶ و ۸ کلمه)
  - AES-128، AES-192 و AES-256
- تعداد دور = ۱۰، ۱۲ و ۱۴
  - هر دور شامل ۴ گام

ساختار AES

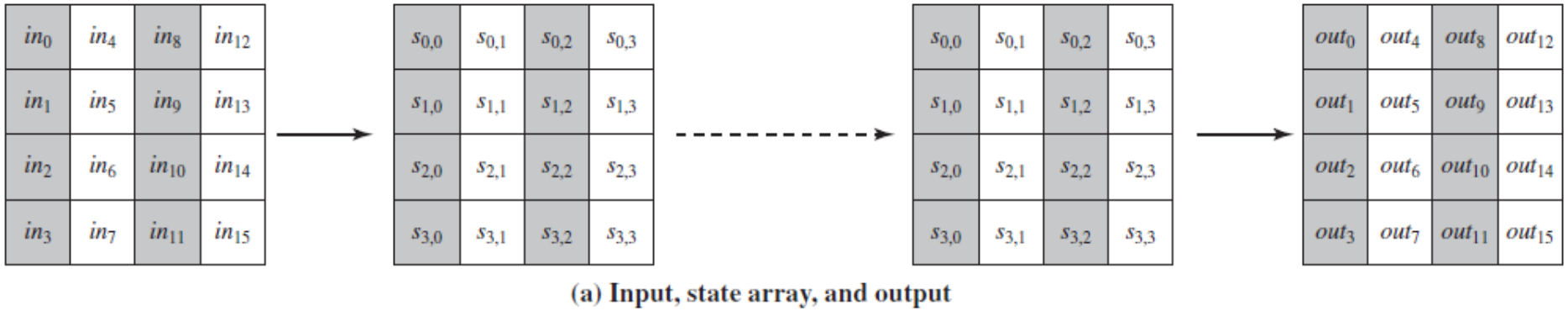
- طول قالب = ۱۲۸ بیت (۱۶ بایت)
  - یک ماتریس ۴ در ۴ از بایت‌ها
  - ماتریس حالت (به صورت ستونی)
- در هر مرحله عملیات بر روی ماتریس حالت صورت می‌گیرد
- کلید
  - کلمه‌ها (۴ بایت) به صورت ستونی
  - $N+1$ کلید دور (ماتریس ۴ در ۴ از بایت‌ها)



No. of rounds	L ((
10	
12	
14	

ساختار AES

- کلید ۱۲۸ بیتی (۱۶ بایت) ← ۴۴ کلمه (۱۷۶ بایت) ← ۱۶ بایت در هر دور (۴ کلمه) ← ماتریس w



(a) Input, state array, and output

(b) Key and expanded key

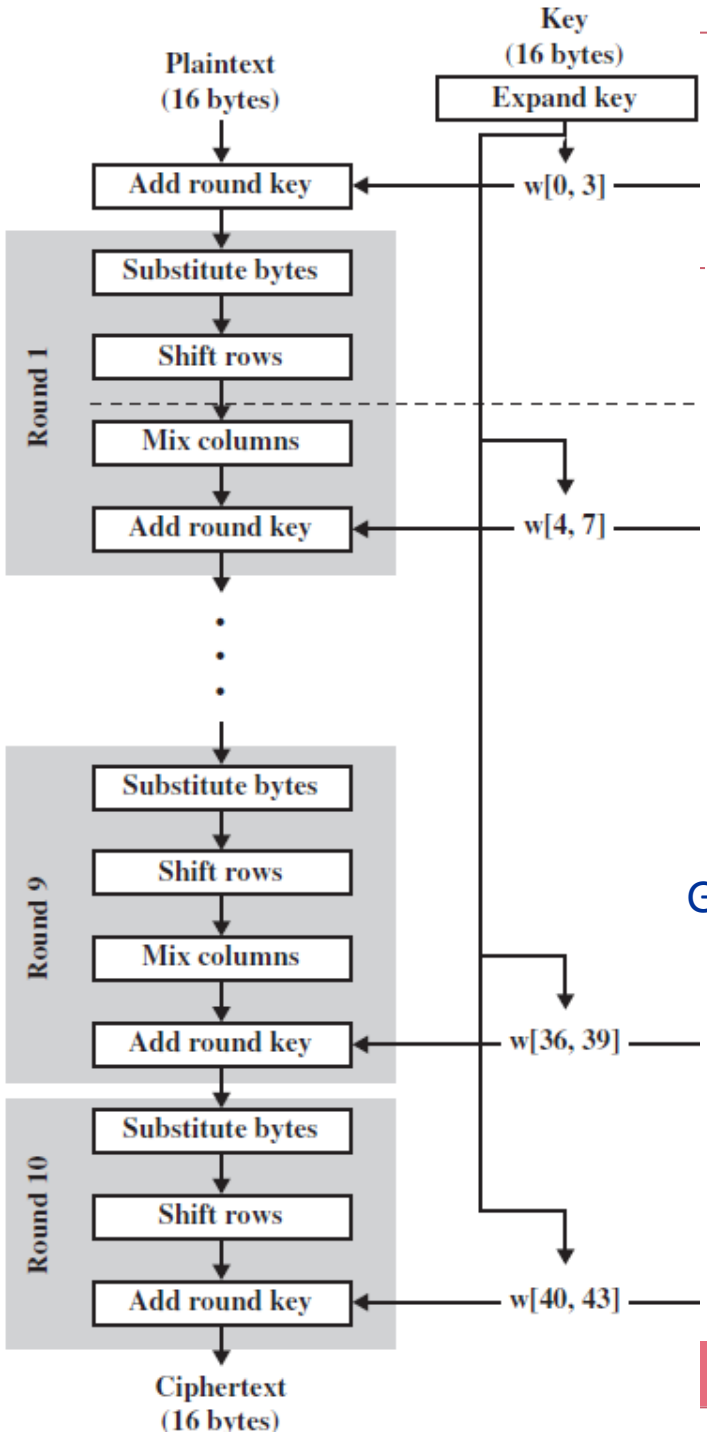
پارامترهای AES

- تعداد دور = N (۱۰، ۱۲، ۱۴) + یک دور اولیه

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

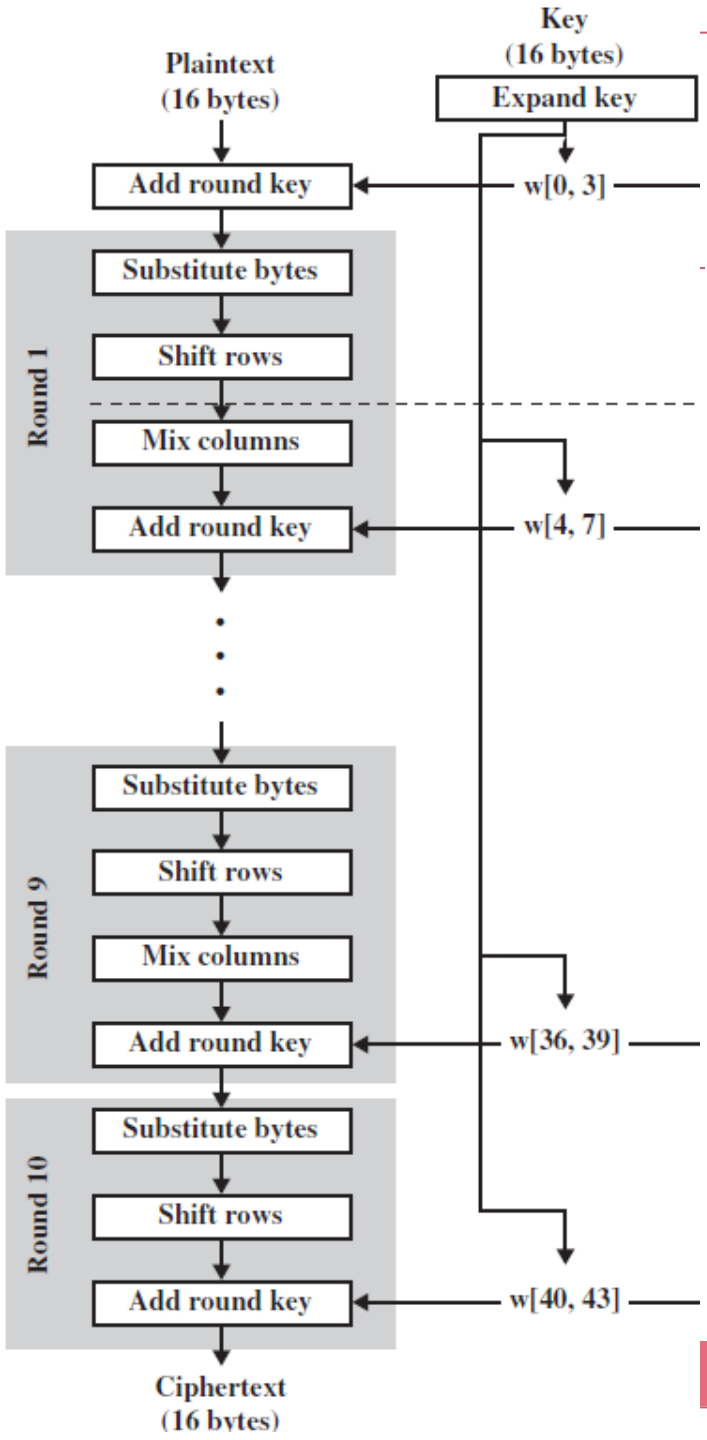
دورهای AES

- دور صفر (اولیه)
  - AddRoundKey: زیرکلید مربوطه اعمال می‌شود
- دور ۱ تا N-1: ۴ گام زیر اعمال می‌شود
  1. **جانشینی بایت‌ها (SubBytes)**: جانشینی هر بایت ماتریس حالت توسط خروجی S-box متناظر
  2. **شیفت سطرها (ShiftRows)**: شیفت سطری بایت‌ها در ماتریس حالت
  3. **ترکیب ستون‌ها (MixColumns)**: ضرب ماتریس حالت در یک ماتریس معین
     - ترکیب خطی ستون‌ها
  4. **کلید دور (AddRoundKey)**: اعمال زیرکلید مربوطه
- دور Nام: گام‌های فوق به جز ترکیب ستون‌ها



رمزگذاری AES

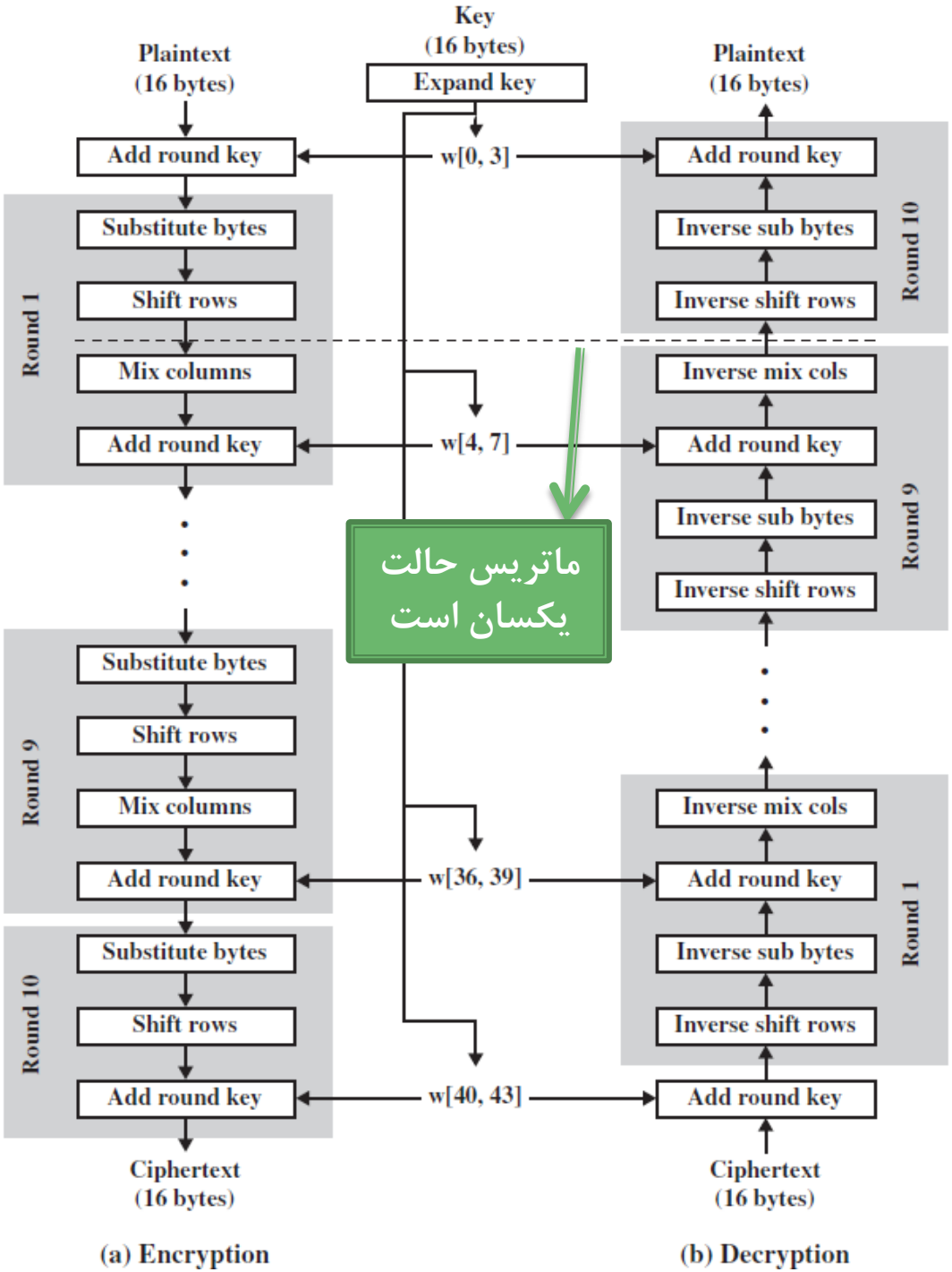
- ساختار فایستلی نیست
  - ورودی ۲ قسمت نمی‌شود و عملیات ماتریسی بایتی است
- کلید اصلی به ۴۴ کلمه ۳۲ بیتی توسیع می‌یابد
  - کلید هر دور: ۴ کلمه (هر کدام ۳۲ بیت)
- هر دور: ۳ عمل جانشینی و یک جایگشت
  - SubBytes: جانشینی بایتی توسط s-box متناظر
  - ShiftRows: جایگشت ساده
  - MixColumns: جانشینی با عملیات بایتی در میدان $GF(2^8)$
  - AddRoundKey: XOR بیتی
- ساختار ساده
- آغاز و پایان الگوریتم با اعمال کلید
  - هر مرحله دیگر برگشت‌پذیر



رمزگذاری AES

- اعمال کلید = مشابه رمز ورنام
- ۳ گام دیگر تواماً
  - confusion، diffusion و غیرخطی بودن
  - فاقد امنیت (کلید ندارند)
  - ترکیب با کلید: امنیت بالا
- معکوس‌پذیر
  - استفاده از نگاشت معکوس در هر گام برای رمزگشایی

رمزگشایی AES

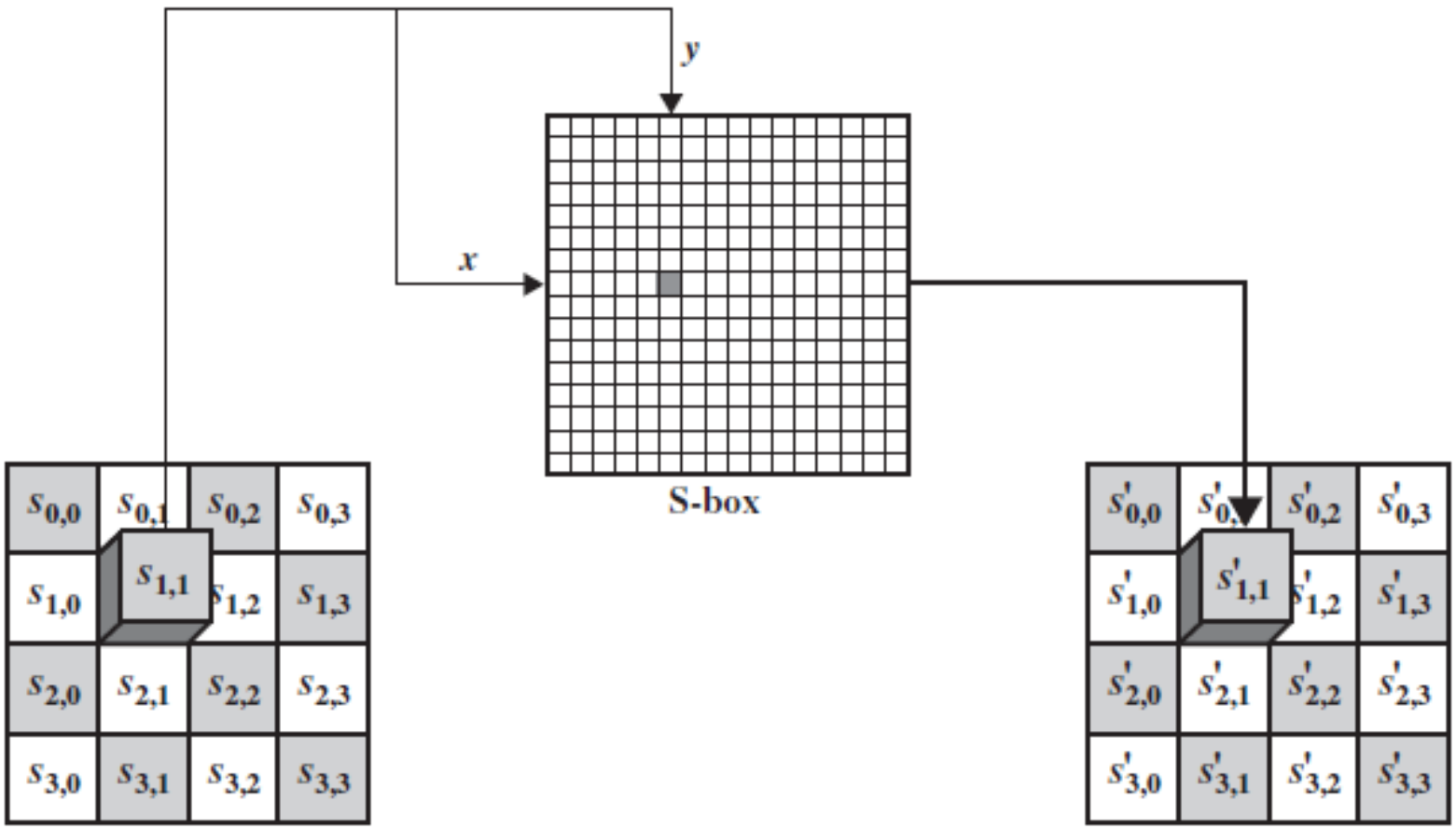


- استفاده از کلید با ترتیب عکس
  - برخلاف قبلی‌ها involution نیست
- ۳ گام در دور پایانی
  - الگوریتم معکوس‌پذیر

جانشینی بایت‌ها (Substitute Bytes Transformation)

- قسمت مستقیم (رمزگذاری) ← forward substitute byte transformation
  - SubBytes
- ماتریس حالت (ماتریس 4×4 از بایت‌ها) به یک ماتریس حالت دیگر نگاشت می‌شود
- نگاشت توسط یک جدول 16×16 با مقادیر بایتی صورت می‌گیرد
  - شامل ۲۵۶ مقدار ممکن برای یک بایت (۸ بیت)
  - اصول ریاضی
- ورودی: هر بایت در ماتریس حالت
  - ۴ بیت چپ: سطر جدول
  - ۴ بیت راست: ستون جدول
  - مقدار جدول = خروجی = جایگزین بایت ورودی در ماتریس حالت می‌شود

جانشینی بایت‌ها



جانشینی بایت‌ها

S-box

- {95}

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	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
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

جانشینی بایت‌ها

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	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
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

S-box

- {95}
- سطر ۹
- ستون ۵
- {2A}

جانشینی بایت‌ها

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

معکوس جانشینی بایت‌ها (رمزگشایی)

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

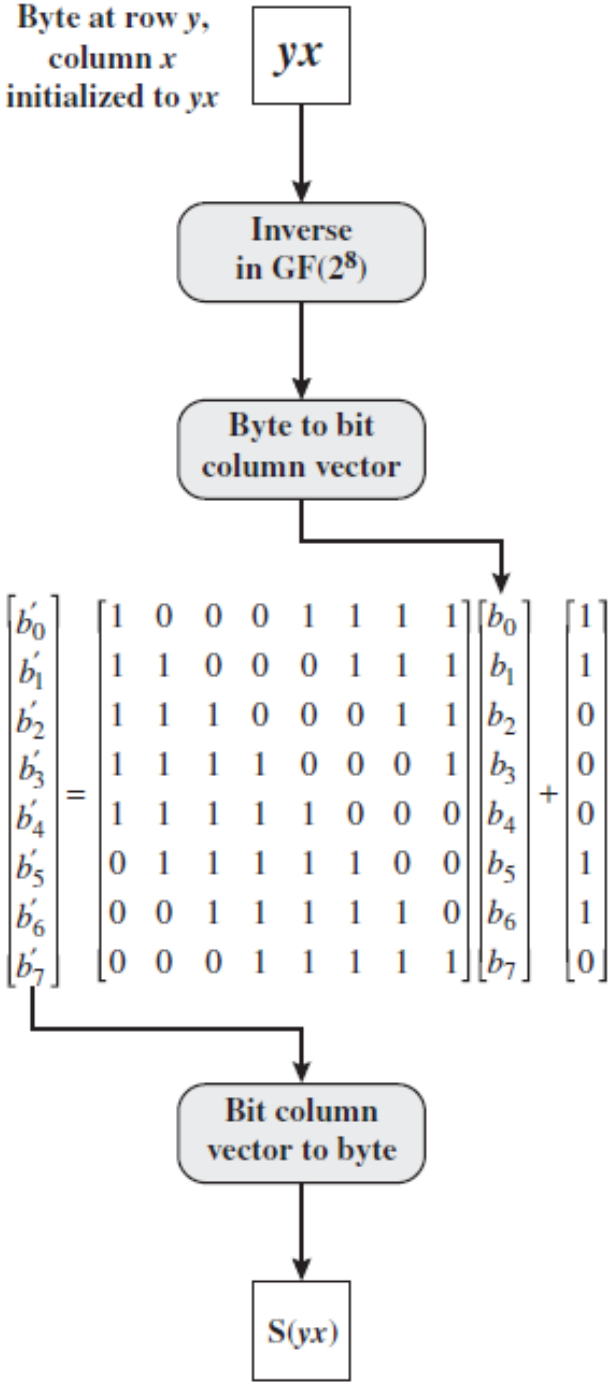
اصول ریاضی جانشینی بایت‌ها

- ضرب ماتریسی + معکوس + جمع

$$x = a^{-1} \quad \left(\mathrm{GF}\left(2^8\right)\right)$$

$$S(a) = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

اصول ریاضی جانشینی بایت‌ها



- معکوس در میدان $GF(2^8)$

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

$$c = \{63\} = (01100011)$$

- جمع در میدان فوق: XOR

اصول ریاضی جانشینی بایت‌ها: مثال

- ورودی: {95}

$$\{95\}^{-1} = \{8A\} = 10001010$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \{2A\}$$

جانشینی بایت‌ها

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	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
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

S-box

- {95}
- سطر ۹
- ستون ۵
- {2A}

معکوس جانشینی بایت‌ها (رمزگشایی)

- **inverse substitute byte transformation**
  - InvSubBytes
  - معکوس عمل فوق
  - ضرب در ماتریس معکوس + جمع برداری + معکوس‌گیری در میدان $GF(2^8)$

$$\{2A\} \rightarrow \{95\}$$

$$b'_i = b_{(i+2) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus d_i$$

$$d = \{05\} = (00000101)$$

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

جانشینی بایت‌ها

- رمزگذاری $\mathbf{B}' = \mathbf{XB} \oplus \mathbf{C}$
- رمزگشایی $\mathbf{Y}(\mathbf{XB} \oplus \mathbf{C}) \oplus \mathbf{D} \stackrel{?}{=} \mathbf{B}$

$$\begin{bmatrix} 0&0&1&0&0&1&0&1\\ 1&0&0&1&0&0&1&0\\ 0&1&0&0&1&0&0&1\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 0&0&1&0&1&0&0&1\\ 1&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0 \end{bmatrix} \begin{bmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{bmatrix} \begin{bmatrix} b_0\\ b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 0&0&1&0&0&1&0&1\\ 1&0&0&1&0&0&1&0\\ 0&1&0&0&1&0&0&1\\ 1&0&1&0&0&1&0&0\\ 0&1&0&1&0&0&1&0\\ 0&0&1&0&1&0&0&1\\ 1&0&0&1&0&1&0&0\\ 0&1&0&0&1&0&1&0 \end{bmatrix} \begin{bmatrix} 1\\ 1\\ 0\\ 0\\ 0\\ 1\\ 1\\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1\\ 0\\ 1\\ 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix} =$$

$$\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} b_0\\ b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7 \end{bmatrix} \oplus \begin{bmatrix} 1\\ 0\\ 1\\ 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1\\ 0\\ 1\\ 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix} = \begin{bmatrix} b_0\\ b_1\\ b_2\\ b_3\\ b_4\\ b_5\\ b_6\\ b_7 \end{bmatrix}$$

$$\mathbf{YX} = \mathbf{I}$$

$$\mathbf{YC} = \mathbf{D}$$

اصول ریاضی جانشینی بایت‌ها

- مقاوم در برابر حملات تحلیل رمز شناخته شده
- هم‌بستگی کم میان بیت‌های ورودی و خروجی
- ساختار غیرخطی
  - معکوس در میدان $GF(2^8)$
- انتخاب مقدار c:
  - S-box نقطه ثابت و ثابت وارون ندارد

$$[\text{S-box}(a) = a] \qquad [\text{S-box}(a) = \overline{a}]$$

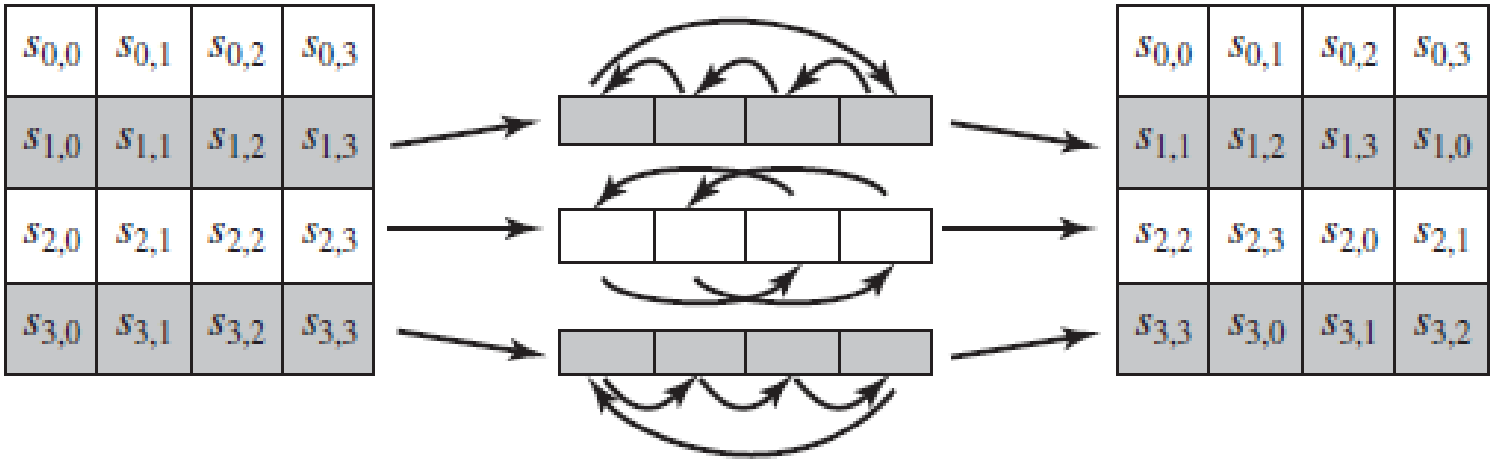
- وارون‌پذیر:

$$\text{IS-box}[\text{S-box}(a)] = a$$

$$\text{S-box}(a) \neq \text{IS-box}(a)$$

شیفت سطری (ShiftRows Transformation)

- تبدیل مستقیم (رمزگذاری): شیفت چرخشی سطرها در ماتریس حالت به چپ
  - سطر اول: صفر بایت (بدون تغییر)
  - سطر دوم: یک بایت
  - سطر سوم: دو بایت
  - سطر سوم: سه بایت



شیفت سطری (ShiftRows Transformation)

- تبدیل مستقیم (رمزگذاری): شیفت چرخشی سطرها در ماتریس حالت به چپ

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

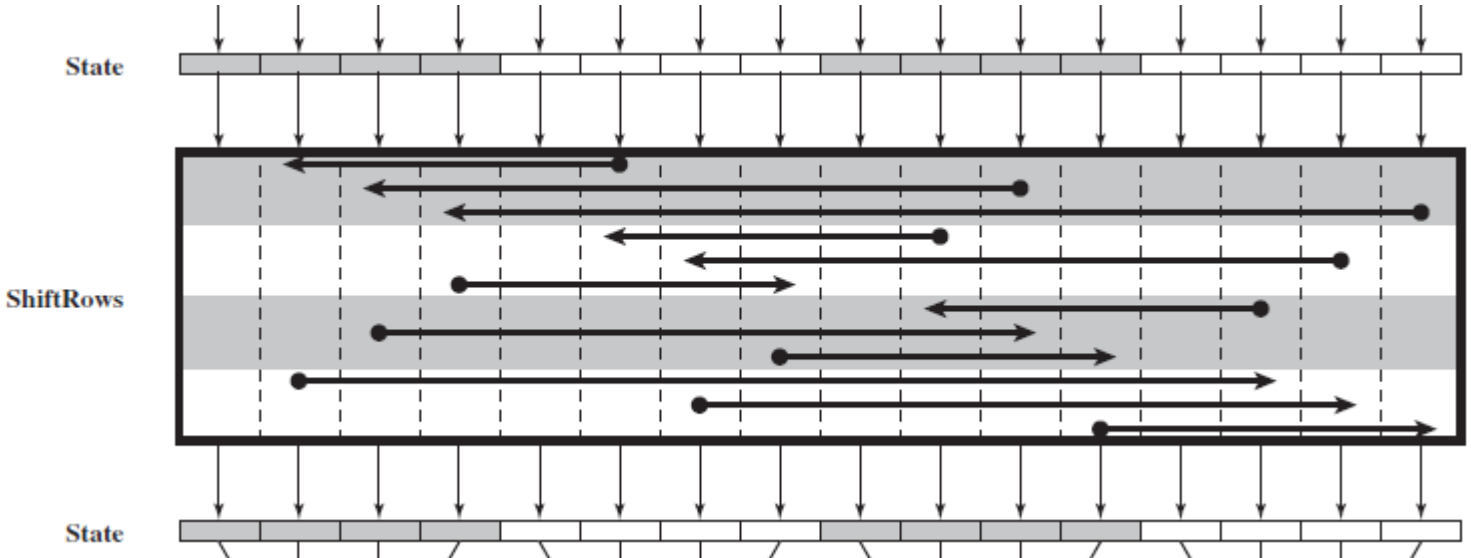
→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

- تبدیل معکوس (رمزگشایی): شیفت چرخشی سطرها در ماتریس حالت به راست

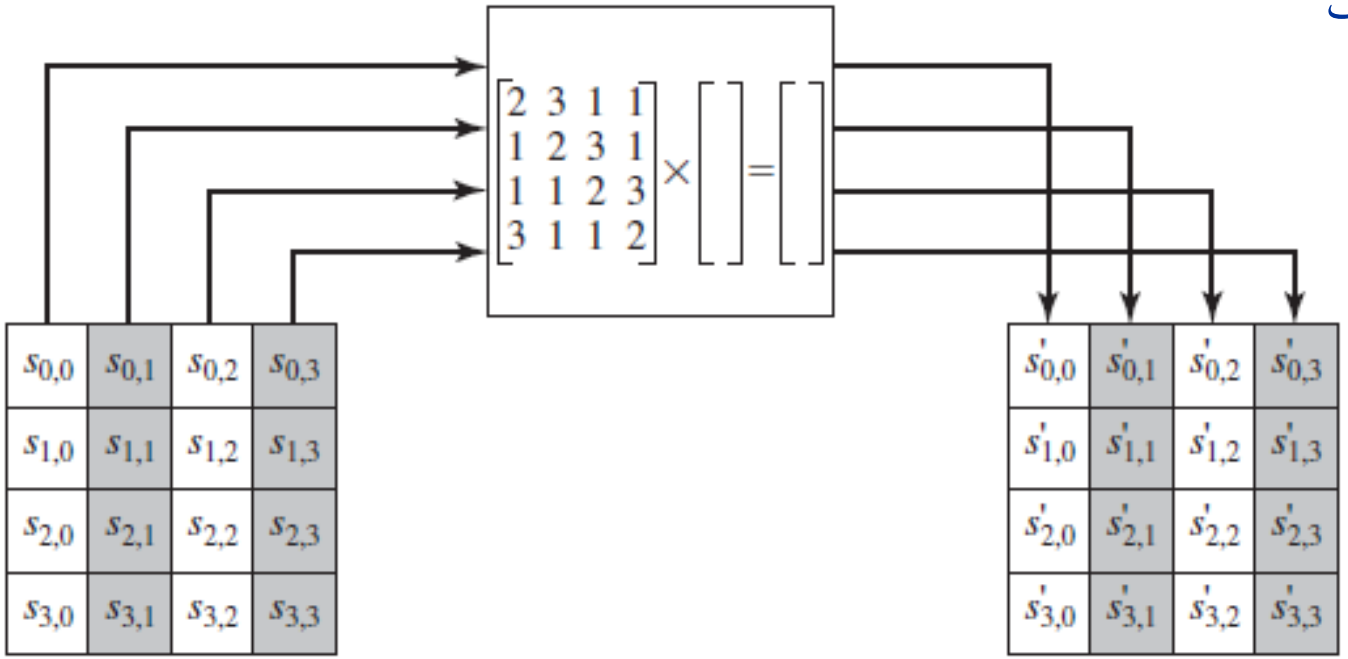
شیفت سطری (ShiftRows Transformation)

- ورودی، خروجی و حالت به صورت ۴ ستون ۴ بایتی هستند
  - ۴ بایت اول متن اصلی در ستون اول قرار می‌گیرند و ...
- اعمال کلید دور به صورت ستونی
- جابجایی سطری بایت‌ها مورد نیاز است
  - شیفت سطری بایت‌ها را از یک ستون به ستون دیگر می‌برد
  - ۴ بایت هر ستون در هر ۴ ستون پخش می‌شوند



ترکیب ستون‌ها (MixColumns Transformation)

- تبدیل مستقیم (رمزگذاری): forward mix column transformation
- ضرب ماتریس حالت در یک ماتریس معین
  - ترکیب خطی ستون‌ها
  - بر هر ستون به طور جداگانه تاثیر می‌گذارد
  - هر بایت در هر ستون با مقدار جدیدی جانشین می‌شود که تابعی از همه ۴ بایت آن ستون است



ترکیب ستون‌ها

- ضرب ماتریسی

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

- ضرب و جمع‌ها در میدان $GF(2^8)$

- برای هرستون:

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

- روش دیگر بیان ترکیب ستون‌ها: چندجمله‌ای

مثال ترکیب ستون‌ها

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

$$(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$
$$\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\}$$
$$\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\}$$
$$(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\}$$

مثال ترکیب ستون‌ها

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

$$(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$
$$\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\}$$
$$\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\}$$
$$(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\}$$

- ضرب در {02} برابر با شیفت به چپ و XOR با (0001 1011) به شرط ۱ بودن LSB

مثال ترکیب ستون‌ها

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

$$\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} = \{47\}$$
$$\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} = \{37\}$$
$$\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) = \{94\}$$
$$(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) = \{ED\}$$

- ضرب در {02} برابر با شیفت به چپ و XOR با (0001 1011) به شرط ۱ بودن MSB

$$\{02\} \cdot \{87\} = (0000\,1110) \oplus (0001\,1011) = (0001\,0101)$$

$$\{03\} \cdot \{6E\} = \{6E\} \oplus \{02\} \cdot \{6E\}$$
$$= (0110\,1110) \oplus (1101\,1100) = (1011\,0010)$$

مثال ترکیب ستون‌ها

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$
$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$
$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$
$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

$$\begin{aligned}
(\{02\} \cdot \{87\}) \oplus (\{03\} \cdot \{6E\}) \oplus \{46\} \oplus \{A6\} &= \{47\} \\
\{87\} \oplus (\{02\} \cdot \{6E\}) \oplus (\{03\} \cdot \{46\}) \oplus \{A6\} &= \{37\} \\
\{87\} \oplus \{6E\} \oplus (\{02\} \cdot \{46\}) \oplus (\{03\} \cdot \{A6\}) &= \{94\} \\
(\{03\} \cdot \{87\}) \oplus \{6E\} \oplus \{46\} \oplus (\{02\} \cdot \{A6\}) &= \{ED\}
\end{aligned}$$

- ضرب در {02} برابر با شیفت به چپ و XOR با (0001 1011) به شرط ۱ بودن LSB

$$\{02\} \cdot \{87\} = (0000\,1110) \oplus (0001\,1011) = (0001\,0101)$$

$$\begin{aligned}
\{03\} \cdot \{6E\} &= \{6E\} \oplus \{02\} \cdot \{6E\} \\
&= (0110\,1110) \oplus (1101\,1100) = (1011\,0010)
\end{aligned}$$

$$\{02\} \cdot \{87\} = (0001\,0101)$$
$$\{03\} \cdot \{6E\} = (1011\,0010)$$
$$\{46\} = (0100\,0110)$$
$$\{A6\} = \underline{(1010\,0110)}$$
$$0100\,0111 = \{47\}$$

ترکیب ستون‌ها (بیان چندجمله‌ای)

- هر ستون ماتریس حالت (۴ بایت) به عنوان یک چندجمله‌ای از درجه ۳ با ضرایب در میدان $GF(2^8)$ است

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

$$col_j(x) = s_{3,j}x^3 + s_{2,j}x^2 + s_{1,j}x + s_{0,j}$$

ترکیب ستون‌ها (بیان چندجمله‌ای)

- هر ستون ماتریس حالت (۴ بایت) به عنوان یک چندجمله‌ای از درجه ۳ با ضرایب در میدان $GF(2^8)$ است

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

$$col_j(x) = s_{3,j}x^3 + s_{2,j}x^2 + s_{1,j}x + s_{0,j}$$

- ضرب ماتریسی برابر با ضرب چندجمله‌ای فوق در چندجمله‌ای زیر است:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

- در پیمانه x^4+1
- عملیات ضرایب در میدان $GF(2^8)$ با چندجمله‌ای $m(x) = x^8 + x^4 + x^3 + x + 1$ است
  - جمع همان XOR است ولی برای ضرب باید چندجمله‌های درجه ۷ محاسبه شوند

ترکیب ستون‌ها (چندجمله‌ای)

- برای یک ستون: $b(x) = col_j(x) = s_{3,j}x^3 + s_{2,j}x^2 + s_{1,j}x + s_{0,j} = b_3x^3 + b_2x^2 + b_1x + b_0$

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$c(x) = a(x) \times b(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$$

- عملیات ضرایب در میدان $GF(2^8)$ با چندجمله‌ای $m(x) = x^8 + x^4 + x^3 + x + 1$

$$c_0 = a_0 \bullet b_0$$
$$c_1 = (a_1 \bullet b_0) \oplus (a_0 \bullet b_1)$$
$$c_2 = (a_2 \bullet b_0) \oplus (a_1 \bullet b_1) \oplus (a_0 \bullet b_2)$$
$$c_3 = (a_3 \bullet b_0) \oplus (a_2 \bullet b_1) \oplus (a_1 \bullet b_2) \oplus (a_0 \bullet b_3)$$
$$c_4 = (a_3 \bullet b_1) \oplus (a_2 \bullet b_2) \oplus (a_1 \bullet b_3)$$
$$c_5 = (a_3 \bullet b_2) \oplus (a_2 \bullet b_3)$$
$$c_6 = a_3 \bullet b_3$$

ترکیب ستون‌ها (چندجمله‌ای)

- برای یک ستون: $b(x) = col_j(x) = s_{3,j}x^3 + s_{2,j}x^2 + s_{1,j}x + s_{0,j} = b_3x^3 + b_2x^2 + b_1x + b_0$

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$c(x) = a(x) \times b(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0$$

- عملیات ضرایب در میدان GF(2^8) با چندجمله‌ای $m(x) = x^8 + x^4 + x^3 + x + 1$

$$c_0 = a_0 \bullet b_0$$
$$c_1 = (a_1 \bullet b_0) \oplus (a_0 \bullet b_1)$$
$$c_2 = (a_2 \bullet b_0) \oplus (a_1 \bullet b_1) \oplus (a_0 \bullet b_2)$$
$$c_3 = (a_3 \bullet b_0) \oplus (a_2 \bullet b_1) \oplus (a_1 \bullet b_2) \oplus (a_0 \bullet b_3)$$
$$c_4 = (a_3 \bullet b_1) \oplus (a_2 \bullet b_2) \oplus (a_1 \bullet b_3)$$
$$c_5 = (a_3 \bullet b_2) \oplus (a_2 \bullet b_3)$$
$$c_6 = a_3 \bullet b_3$$

$$\begin{aligned} d(x) &= c(x) \bmod (x^4 + 1) \\ &= [c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0] \bmod (x^4 + 1) \\ &= c_3x^3 + (c_2 \oplus c_6)x^2 + (c_1 \oplus c_5)x + (c_0 \oplus c_4) \end{aligned}$$

ترکیب ستون‌ها

- ضرایب چندجمله‌ای $a(x)$ (یا ضرایب ماتریس) از یک کد خطی با فاصله بیشینه میان کلمات کد انتخاب شده‌اند
  - بایت‌ها در هر ستون با بیشترین فاصله از هم جابجا می‌شوند
  - انتخاب مقادیر {01}، {02} و {03} با توجه به پیاده‌سازی ساده آن‌ها صورت گرفته است
    - ضرب در {02} برابر با شیفت به چپ و XOR شرطی با (0001 1011)
- اعمال شیفت سطری و ترکیب ستون‌ها
  - پس از چند دور، تمامی بیت‌های خروجی وابسته به تمامی بیت‌های ورودی هستند

ترکیب ستون‌ها

- تبدیل معکوس (رمزگشایی): inverse mix column transformation
- ضرب در ماتریس معکوس

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ضرب چندجمله‌ای ستون در چندجمله‌ای زیر

$$b(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$$

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

$$b(x) = a^{-1}(x) \bmod (x^4 + 1)$$

ترکیب ستون‌ها

- تبدیل معکوس (رمزگشایی)
  - پیاده‌سازی پیچیده‌تر

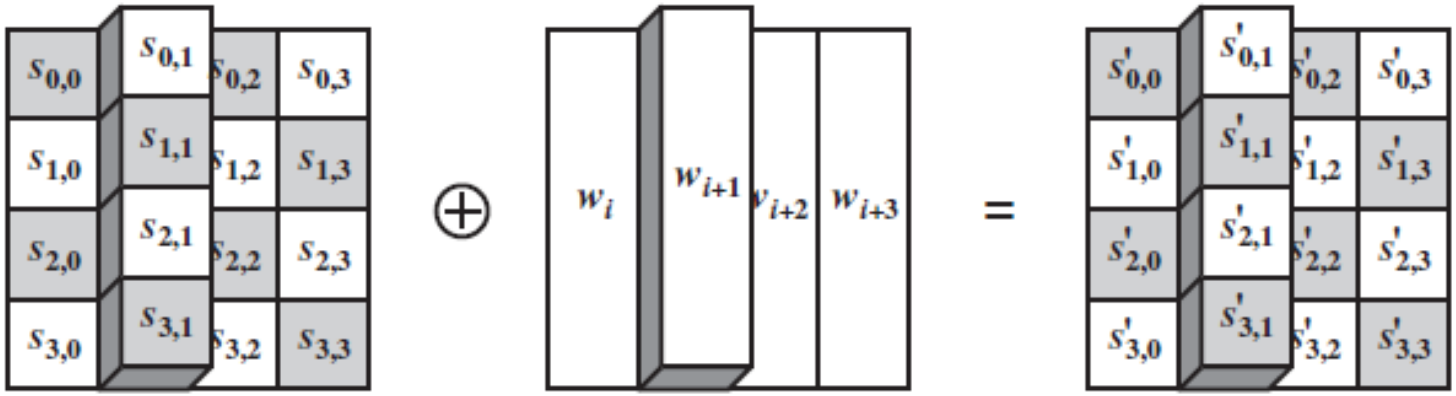
$$b(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}$$

- در بیشتر موارد تنها رمزگذاری مورد نیاز است

- در مودهای CFB و OFB تنها از رمزگذاری استفاده می‌شود
- در کدهای احراز اصالت تنها از رمزگذاری استفاده می‌شود

اعمال کلید دور (AddRoundKey Transformation)

- XOR بیتی ماتریس حالت با کلید دور (۱۲۸ بیت)
  - به صورت ستونی میان ۴ بایت حالت و ۱ کلمه کلید



- معکوس نیز مشابه مستقیم است

اعمال کلید دور

- ساده‌ترین حالت اعمال کلید
- بر هر بیت تاثیر می‌گذارد
- امنیت
  - پیچیدگی بسط کلید (تولید کلید دور)
  - گام‌های دیگر AES

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

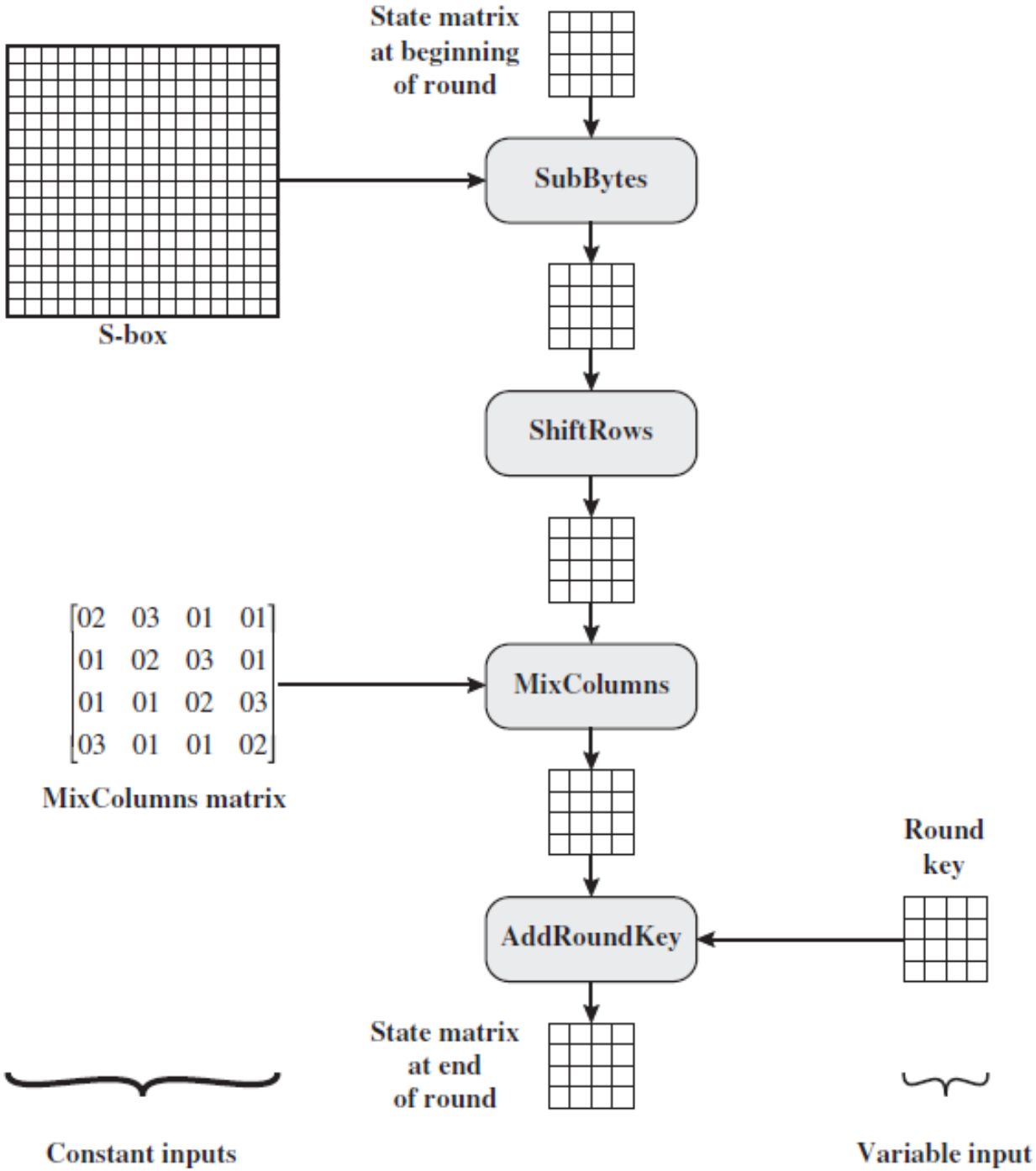
$\oplus$

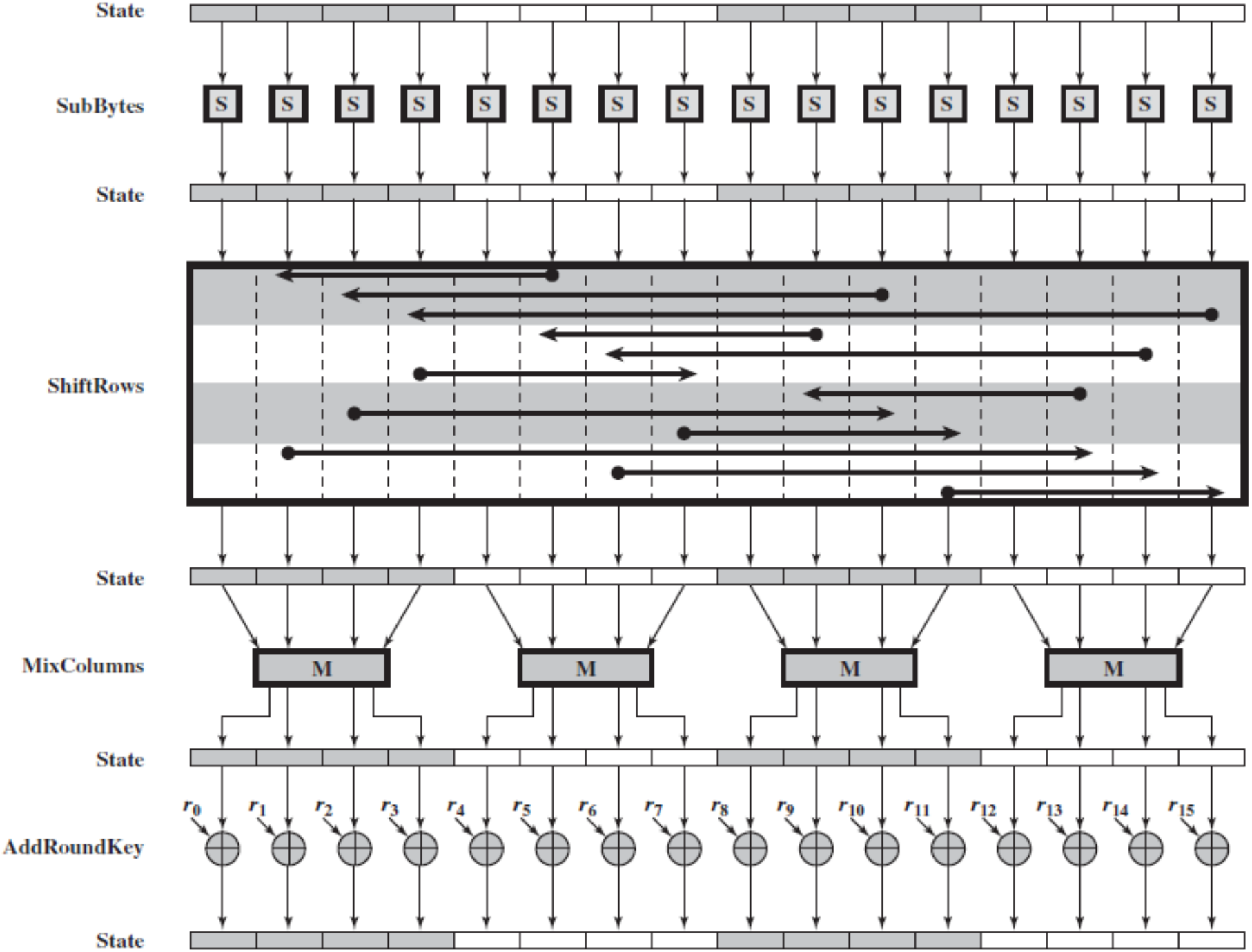
AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

یک دور AES


State matrix at beginning of round
SubBytes
S-box
ShiftRows
MixColumns
02 03 01 01
01 02 03 01
01 01 02 03
03 01 01 02
MixColumns matrix
Round key
AddRoundKey
State matrix at end of round
Constant inputs
Variable input


State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

بسط کلید AES

- ورودی: کلید ۱۲۸ بیتی (۱۶ بایت – ۴ کلمه) ← خروجی: کلید بسط یافته: ۴۴ کلمه (۱۷۶ بایت) ← ۱۶ بایت در هر دور (۴ کلمه) ← ۱+۱۰ دور ← ماتریس ***w***

- ابتدا، کلید در ۴ کلمه اول کلید بسط یافته کپی می‌شود
- سپس، در هر گام ۴ کلمه دیگر از کلید بسط یافته تولید (پُر) می‌شود
  - هر کلمه جدید ***w***[*i*] به کلمه قبلی ***w***[*i*-1] و ۴ کلمه قبل‌تر ***w***[*i*-4] بستگی دارد
  - در ۳ تا از این ۴ کلمه، دو مقدار فوق XOR می‌شوند
  - در کلمات مضارب ۴ در ***w***، تابع پیچیده‌تری بکار می‌رود

بسط کلید AES

- If *i*=4*k*: $w[i] = \text{SubWord}\left(\text{RotWord}\left(w[i-1]\right)\right) \oplus \text{Rcon}\left[\frac{i}{4}\right] \oplus w[i-4]$
- Otherwise: $w[i] = w[i-1] \oplus w[i-4]$

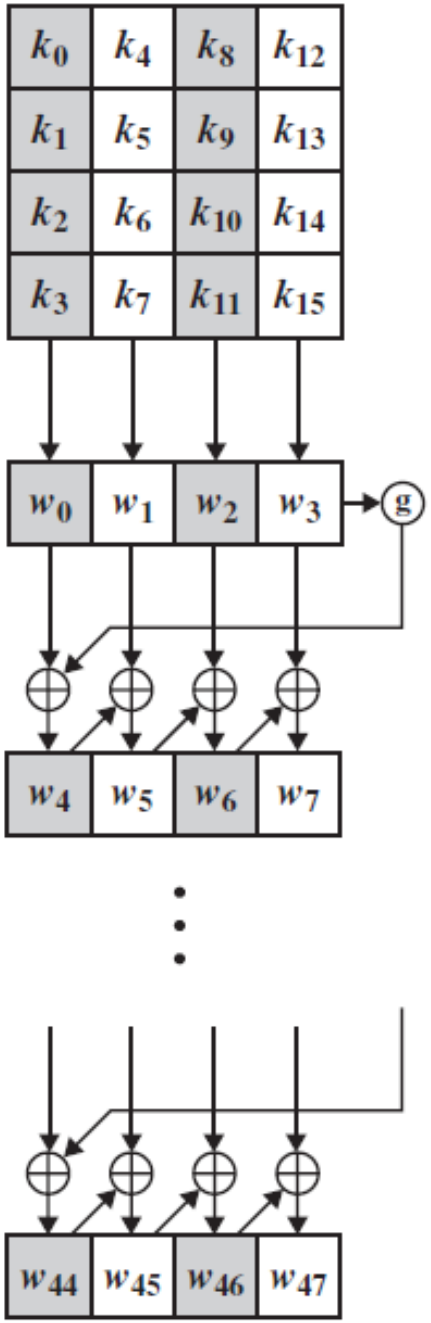
```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);
    for (i = 4; i < 44; i++)
    {
     temp = w[i - 1];
     if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                ⊕ Rcon[i/4];
     w[i] = w[i-4] ⊕ temp
    }
}
```

بسط کلید AES

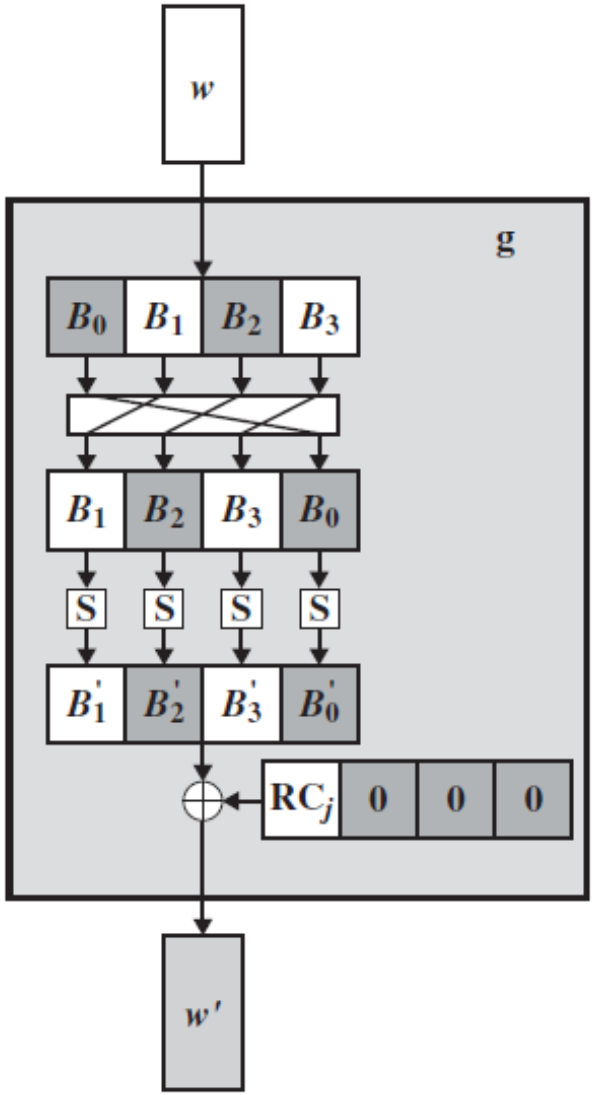
- If $i=4k$:

$$w[i] = \text{SubWord}\left(\text{RotWord}\left(w[i-1]\right)\right) \oplus \text{Rcon}\left[\frac{i}{4}\right] \oplus w[i-4]$$

- Otherwise: $w[i] = w[i-1] \oplus w[i-4]$



تابع g در بسط کلید AES



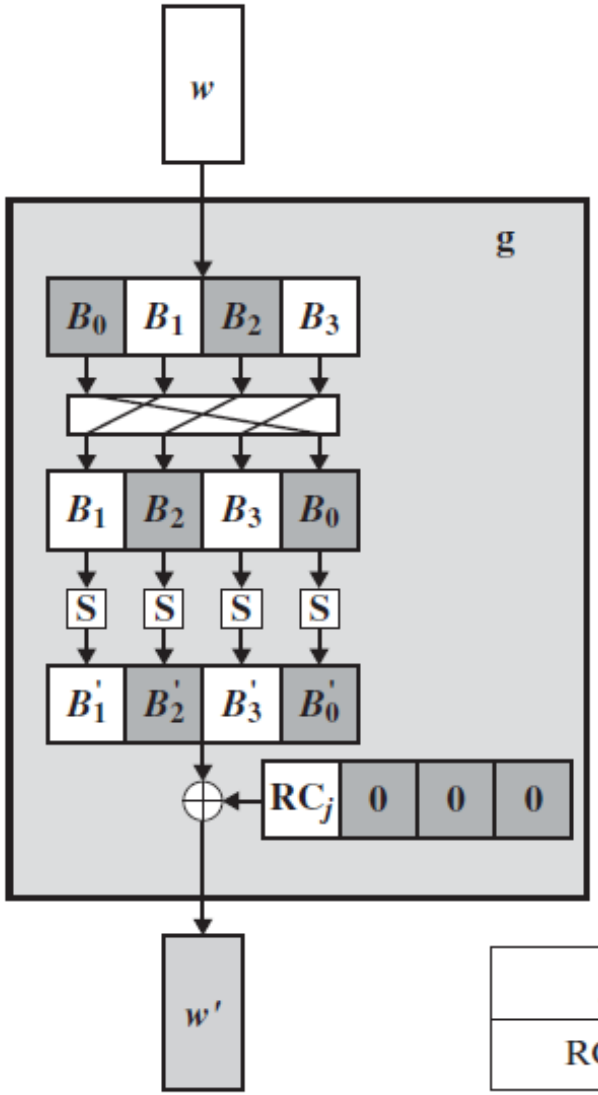
$$\text{SubWord}\left(\text{RotWord}\left(w[i-1]\right)\right) \oplus \text{Rcon}\left[\frac{i}{4}\right]$$

- RotWord: شیفت چرخشی یک بایت به چپ

$[B_0, B_1, B_2, B_3] \longrightarrow [B_1, B_2, B_3, B_0]$

- SubWord: جانشینی بایتی بر اساس s-box قبلی

تابع g در بسط کلید AES



$$\mathrm{SubWord}\left(\mathrm{RotWord}\left(w[i-1]\right)\right) \oplus \mathrm{Rcon}\left[\frac{i}{4}\right]$$

- RotWord: شیفت چرخشی یک بایت به چپ

$[B_0, B_1, B_2, B_3] \longrightarrow [B_1, B_2, B_3, B_0]$

- SubWord: جانشینی بایتی بر اساس s-box قبلی
- Rcon[j]: مقدار معین زیر (ثابت دور)

$$\mathrm{Rcon}[j] = (\mathrm{RC}[j], 0, 0, 0)$$

$$\mathrm{RC}[1] = 1,\ \mathrm{RC}[j] = 2 \bullet \mathrm{RC}[j-1]$$

- عملیات در میدان $GF(2^8)$

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

بسط کلید AES

- طراحی: مقابله با حملات شناخته شده
- ثابت دور: از بین بردن تقارن در تولید کلید هر دور
- با دانستن (قسمتی از) کلید یک دور، اطلاعات زیادی در مورد کلید دورهای دیگر بدست نیاید
- سرعت عملیات
- پراکنش کلید اصلی در کلیدهای دور
  - هر بیت کلید اصلی بر تعداد زیادی از بیت‌های کلیدهای دور تاثیر گذارد
- غیرخطی بودن
  - از تفاضل کلیدهای دور نتوان به تفاضل کلید اصلی رسید
- بیان ساده

اثر بهمنی (Avalanche Effect)

- تغییر یک بیت ورودی (یا کلید)، تعداد زیادی از بیت‌های خروجی را تغییر دهد

Table 5.5 Avalanche Effect in AES: Change in Plaintext

Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210	1
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c	20
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5	58
3	7115262448dc747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62	59
4	f867aee8b437a5210c24c1974cffeabc 43efdb697244df808e8d9364ee0ae6f5	61
5	721eb200ba06206dcbd4bce704fa654e 7b28a5d5ed643287e006c099bb375302	68
6	0ad9d85689f9f77bc1c5f71185e5fb14 3bc2d8b6798d8ac4fe36a1d891ac181a	64
7	db18a8ffa16d30d5f88b08d777ba4eaa 9fb8b5452023c70280e5c4bb9e555a4b	67
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40	65
9	cca104a13e678500ff59025f3bafaa34 b56a0341b2290ba7dfdfbddcd8578205	61
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cde116227ce72433f0	58

اثر بهمنی (Avalanche Effect)

- تغییر یک بیت ورودی (یا کلید)، تعداد زیادی از بیت‌های خروجی را تغییر دهد

Table 5.5 Avalanche Effect in AES: Change in Plaintext

Round	
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5
3	7115262448dc747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40
9	cca104a13e678500ff59025f3bafaa34 b56a0341b2290ba7dfdfbddcd8578205
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cde116227ce72433f0

Table 5.6 Avalanche Effect in AES: Change in Key

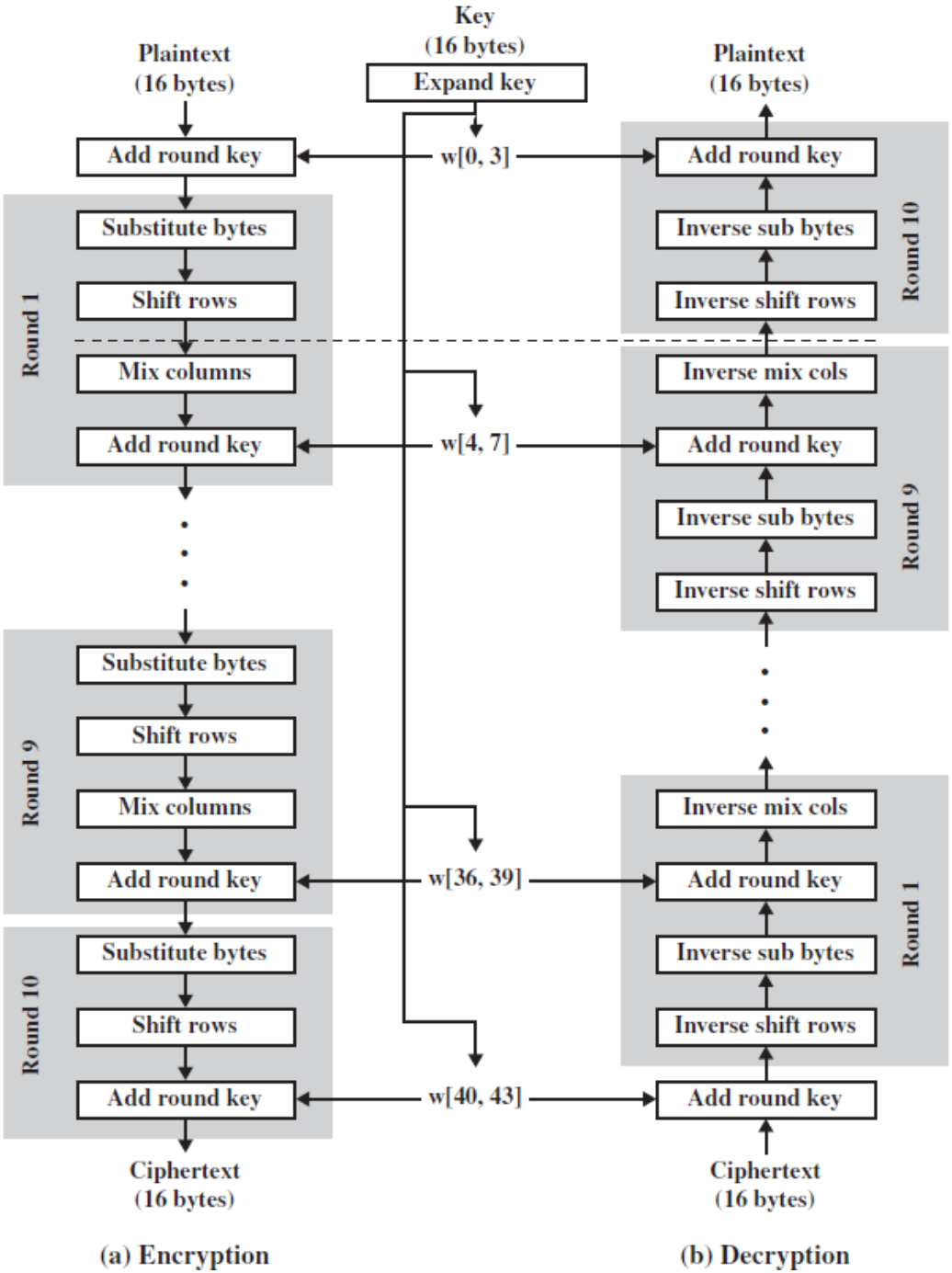
Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0123456789abcdeffedcba9876543210	0
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c5a9ad090ec7ff3fc1e8e8ca4cd02a9c	22
2	5c7bb49a6b72349b05a2317ff46d1294 90905fa9563356d15f3760f3b8259985	58
3	7115262448dc747e5cdac7227da9bd9c 18aeb7aa794b3b66629448d575c7cebf	67
	f867aee8b437a5210c24c1974cffeabc f81015f993c978a876ae017cb49e7eec	63
	721eb200ba06206dcbd4bce704fa654e 5955c91b4e769f3cb4a94768e98d5267	81
	0ad9d85689f9f77bc1c5f71185e5fb14 dc60a24d137662181e45b8d3726b2920	70
	db18a8ffa16d30d5f88b08d777ba4eaa fe8343b8f88bef66cab7e977d005a03c	74
8	f91b4fbfe934c9bf8f2f85812b084989 da7dad581d1725c5b72fa0f9d9d1366a	67
9	cca104a13e678500ff59025f3bafaa34 0ccb4c66bbfd912f4b511d72996345e0	59
10	ff0b844a0853bf7c6934ab4364148fb9 fc8923ee501a7d207ab670686839996b	53

0f1571c947d9e8590cb7add6af7f6798

0e1571c947d9e8590cb7add6af7f6798

پیاده‌سازی AES

- رمزگشایی نیاز به پیاده‌سازی جداگانه دارد
  - همان بسط کلید
- رمزگشایی **معادل** با همان ساختار رمزگذاری
  - بسط کلید متفاوت



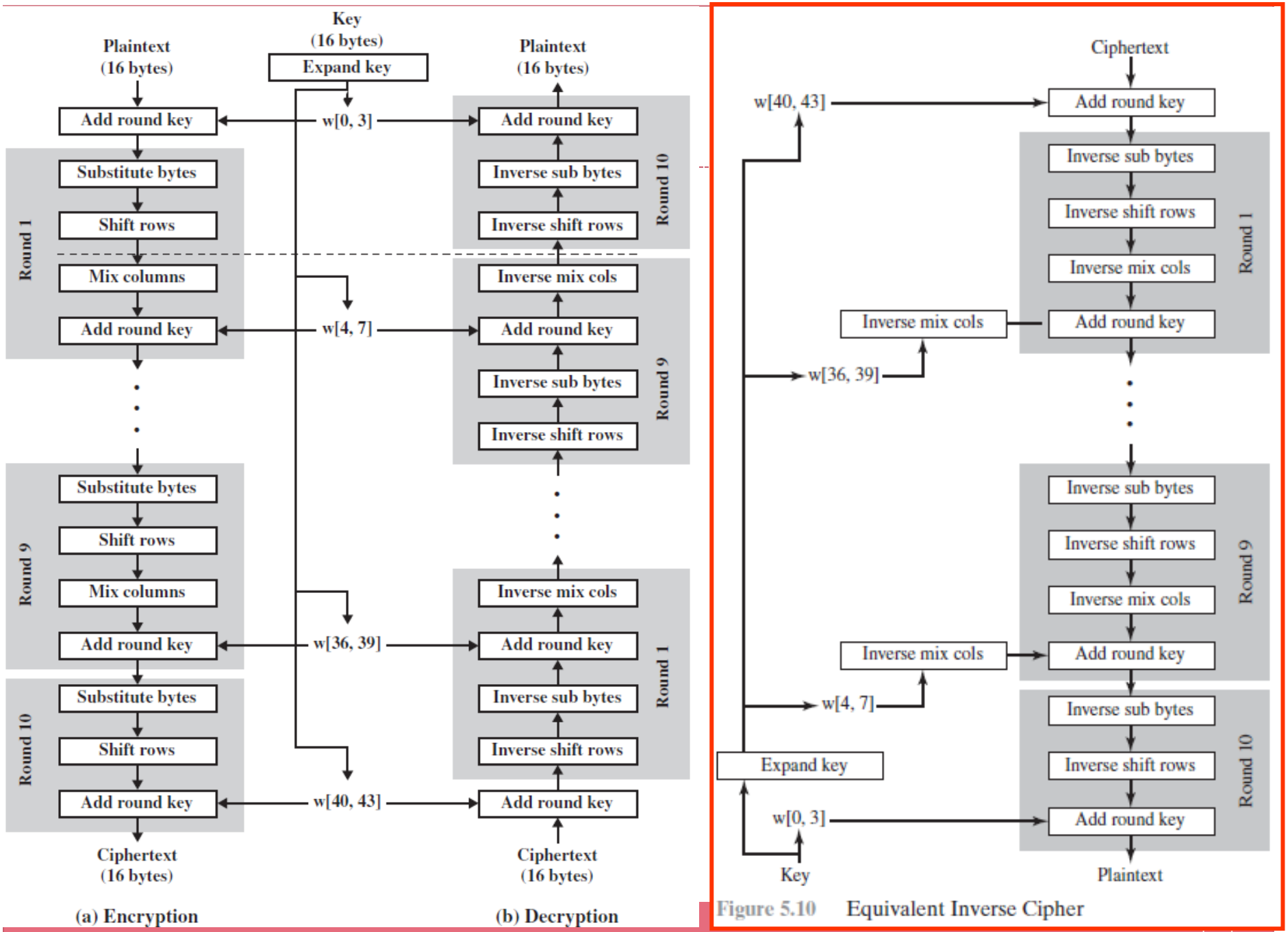


Figure 5.10 Equivalent Inverse Cipher

پیاده‌سازی AES

- قابل پیاده‌سازی کارآ در پردازنده‌های ۸ بیتی
  - کارت‌های هوشمند
  - عملیات بایتی: شیفت + XOR (شرطی) + جدول (۲۵۶ بایت)
  - ضرب در {02} ممکن است منجر به حمله زمانی گردد ← استفاده از جدول
- قابل پیاده‌سازی کارآ در پردازنده‌های ۳۲ بیتی
  - PC
  - با تعریف عملیات بر روی کلمات ۳۲ بیتی کارآتر است
- طراحان Rijndael بر این باورند که پیاده‌سازی فشرده و کارآی این الگوریتم موجب انتخاب آن برای AES شده است

الگوریتم‌های معروف رمز قالبی

- **IDEA** (1990)
- **Blowfish** (1994)
- **RC5** (1994)
- **CAST-128** (1997)

سبک‌های کاری رمزهای قالبی

- رمزهای قالبی با اعمال یک کلید بر قالب b بیتی از ورودی، خروجی b بیتی را تولید می‌کنند
- اگر طول دنباله متن اصلی بیشتر از b بیت باشد، می‌توان آن را به قالب‌های b بیتی تقسیم کرد
- اگر تعداد زیادی از قالب‌ها با یک کلید رمز شوند، امنیت کاهش می‌یابد

- **سبک (mode) کاری** روشی برای افزایش تاثیر (امنیت) یک الگوریتم رمزنگاری و یا سازگار کردن آن با یک کاربرد خاص می‌باشد

سبک‌های کاری رمزهای قالبی

- ۵ سبک کاری مهم (تعریف شده توسط NIST در استاندارد SP 800-38A)
  - هر یک از رمزهای قالبی می‌توانند در هر یک از سبک‌های زیر بکار گرفته شوند
  - کاربردهای متفاوت رمز قالبی را پوشش می‌دهند

1. سبک کتابچه رمز (electronic codebook mode)
2. سبک زنجیره‌ای قالب‌های رمز (cipher block chaining mode)
3. سبک بازخورد رمز (cipher feedback mode)
4. سبک بازخورد خروجی (output feedback mode)
5. سبک شمارنده (counter mode)

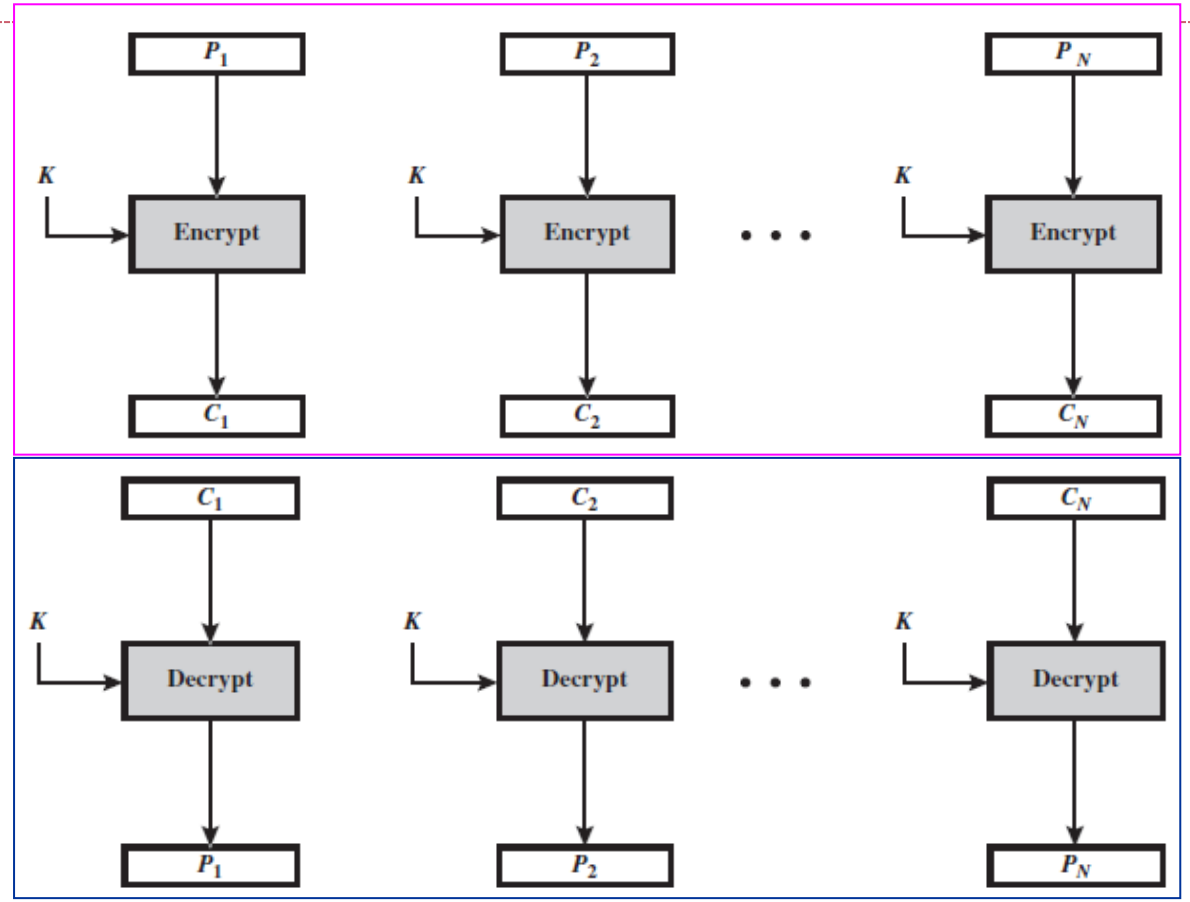
سبک‌های کاری رمزهای قالبی

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

سبک کتابچه رمز (electronic codebook mode) ECB

- ساده‌ترین سبک
- در هر بار یک قالب متن اصلی رمز می‌شود
- برای تمامی قالب‌ها از یک کلید استفاده می‌شود
- کتابچه رمز: برای کلید معین، متن رمز شده برای هر قالب b بیتی متن اصلی یکتا است
- اگر طول دنباله متن اصلی بیشتر از b بیت باشد، آن را به قالب‌های b بیتی تقسیم می‌کنیم
  - دنباله‌زدن (padding) قالب نهایی در صورت نیاز
- رمزگشایی: هر بار یک قالب با کلید یکسان

سبک کتابچه رمز (electronic codebook mode)



- رمزگذاری
- رمزگشایی

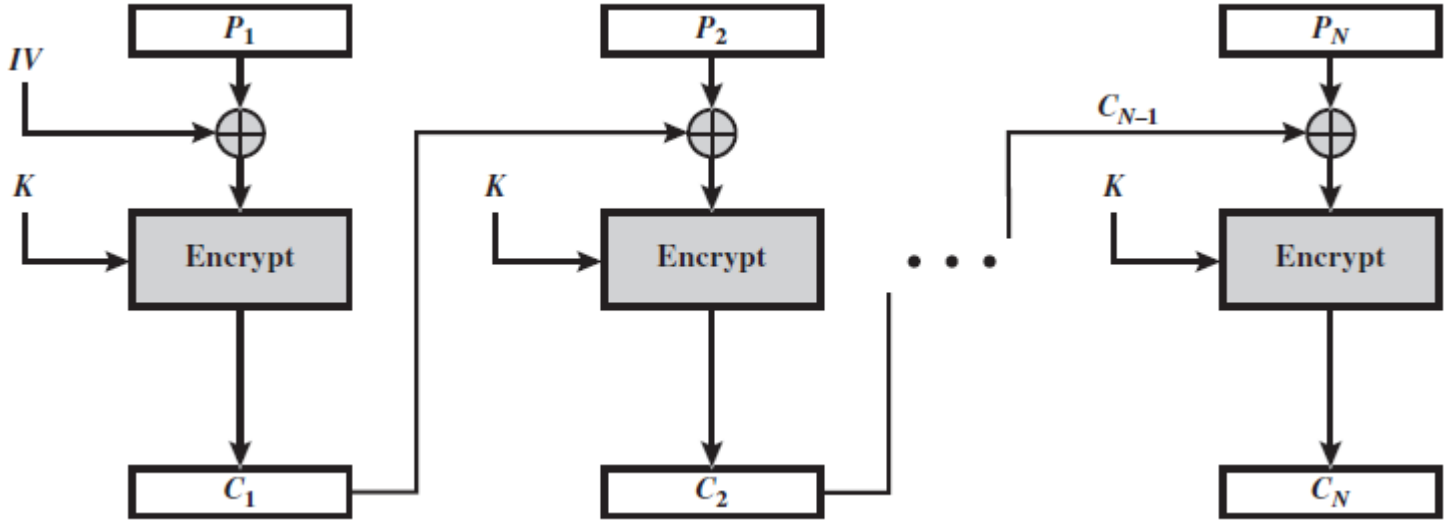
ECB	$C_j = \mathrm{E}(K, P_j)$ $j = 1, \ldots, N$	$P_j = \mathrm{D}(K, C_j)$ $j = 1, \ldots, N$

سبک کتابچه رمز (electronic codebook mode)

- مناسب برای متن اصلی کوتاه
  - کلید AES یا DES
- اگر قالب b بیتی در متن اصلی بیش از یک بار تکرار شود، متن رمز شده یکسان است
- متن اصلی طولانی: ممکن است امن نباشد
  - اگر ساختار متن اصلی مشخص باشد ← حمله تحلیلی
  - شروع و پایان با مقادیر مشخص: تعدادی زوج متن اصلی- متن رمز شده معلوم
  - برخی بیت‌ها به طور متناوب (با مضربی از b) تکرار شوند
- حملات ممکن: تحلیل رمز، تغییر یا جابجایی قالب‌ها

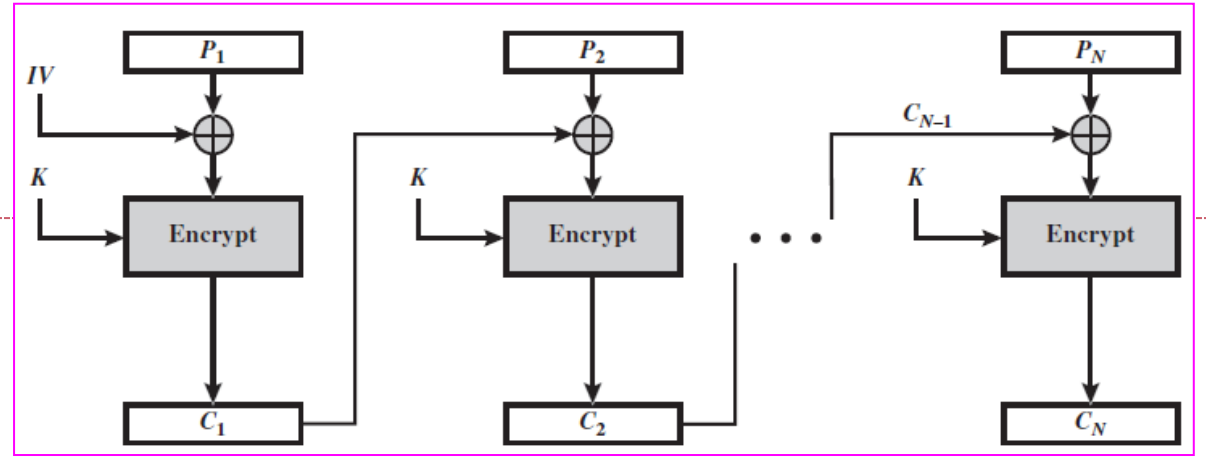
سبک زنجیره‌ای قالب‌های رمز (cipher block chaining mode) CBC

- برای غلبه بر مشکلات ECB
  - اگر قالب b بیتی در متن اصلی بیش از یک بار تکرار شود، متن رمز شده یکسان نباشد
- رمزگذاری

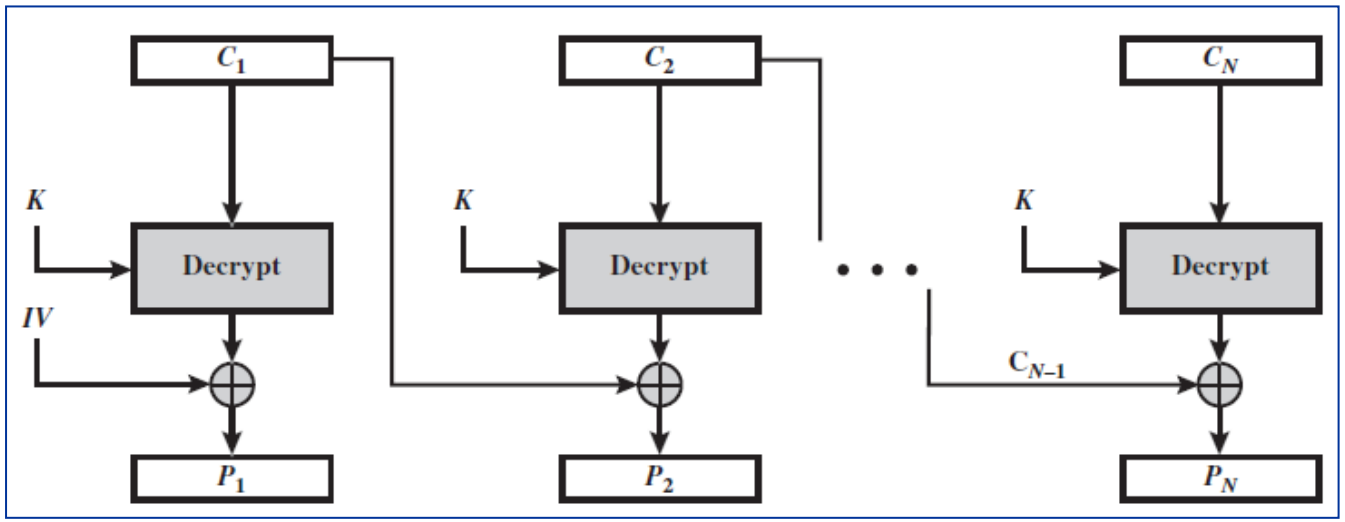


  - ورودی هر بار الگوریتم، رابطه ثابتی با متن اصلی ندارد
  - initialization vector (IV)

CBC



- رمزگذاری



- رمزگشایی

$$C_j = \mathrm{E}(K, [C_{j-1} \oplus P_j])$$

$$\mathrm{D}(K, C_j) = \mathrm{D}(K, E(K, [C_{j-1} \oplus P_j]))$$
$$\mathrm{D}(K, C_j) = C_{j-1} \oplus P_j$$
$$C_{j-1} \oplus \mathrm{D}(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

CBC

CBC	$C_1 = \mathrm{E}(K, [P_1 \oplus \mathrm{IV}])$ $C_j = \mathrm{E}(K, [P_j \oplus C_{j-1}])\ \ j = 2, \ldots, N$	$P_1 = \mathrm{D}(K, C_1) \oplus \mathrm{IV}$ $P_j = \mathrm{D}(K, C_j) \oplus C_{j-1}\ \ \ j = 2, \ldots, N$

- مقدار IV باید تنها برای فرستنده و گیرنده مشخص باشد
  - برای هر متن اصلی
  - می‌توان از سبک ECB استفاده کرد
- اگر IV به طور آشکار ارسال شود، ممکن است دشمن بتواند مقدار دیگری را جایگزین آن کند و قالب اول متن آشکار شده متفاوت با متن اصلی شود

$$C_1 = \mathrm{E}(K, [\mathrm{IV} \oplus P_1])$$
$$P_1 = \mathrm{IV} \oplus \mathrm{D}(K, C_1)$$

$$P_1[i] = \mathrm{IV}[i] \oplus \mathrm{D}(K, C_1)[i]$$

$$P_1[i]' = \mathrm{IV}[i]' \oplus \mathrm{D}(K, C_1)[i]$$

الزامات CBC

- مقدار IV غیرقابل پیش‌بینی باشد
- ۲ روش در Sp800-38a:

1. اعمال رمزنگاری به یک تک‌شمار (nonce) با استفاده از همان کلید الگوریتم اصلی
   - تک‌شمار برای هر بار اجرای سبک، یکتا است مانند یک شمارنده، شماره پیام و ...
2. استفاده از مولد اعداد (شبه)تصادفی

- رمزگذاری قابل موازی‌سازی نیست
- رمزگشایی قابل موازی‌سازی است

- مناسب برای رمزنگاری با طول بالای متن اصلی
  - محرمانگی و احراز اصالت

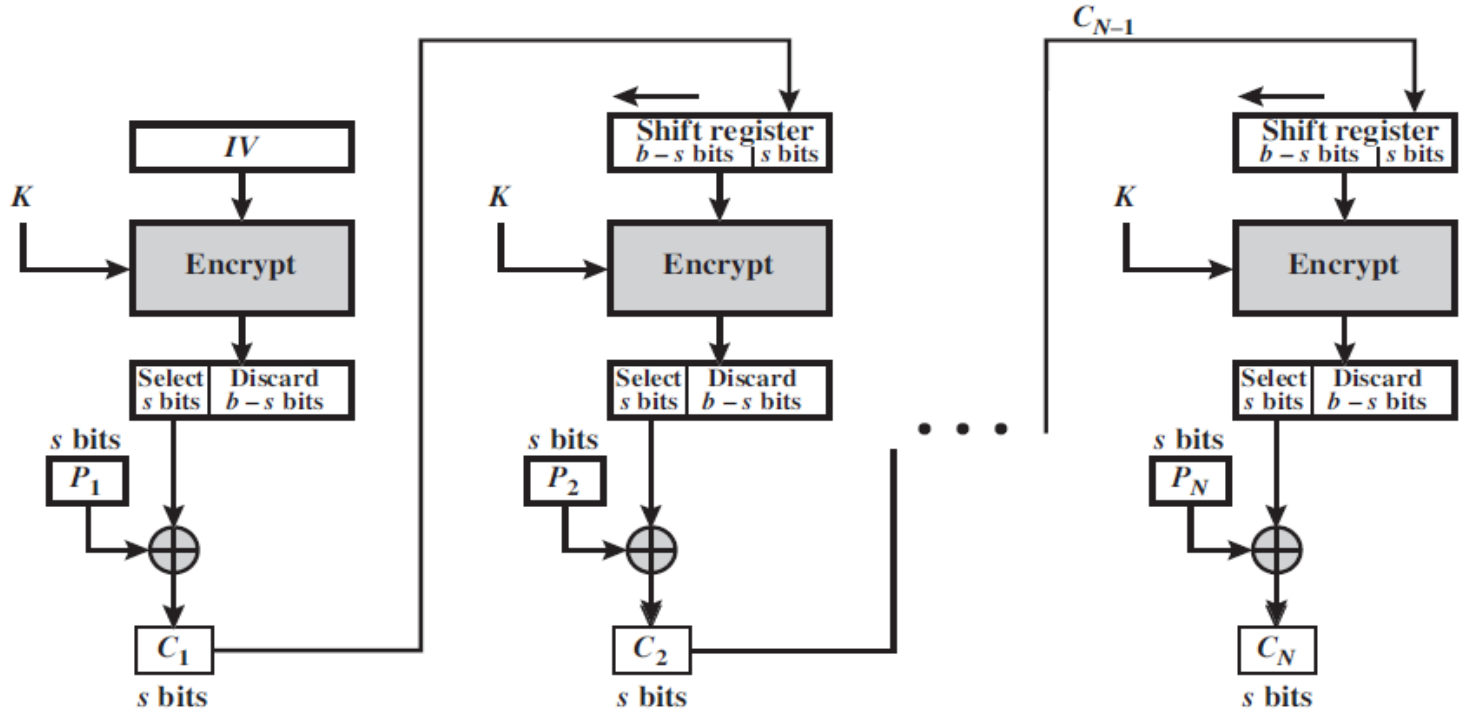
سبک بازخورد رمز (cipher feedback mode)
CFB

- با استفاده از سه سبک زیر می‌توان رمز قالبی را به دنباله‌ای تبدیل کرد:
  - سبک بازخورد رمز (cipher feedback mode)
  - سبک بازخورد خروجی (output feedback mode)
  - سبک شمارنده (counter mode)

- مشکل دنباله‌زدن (padding) برطرف می‌شود
  - طول دنباله ورودی و خروجی برابر است

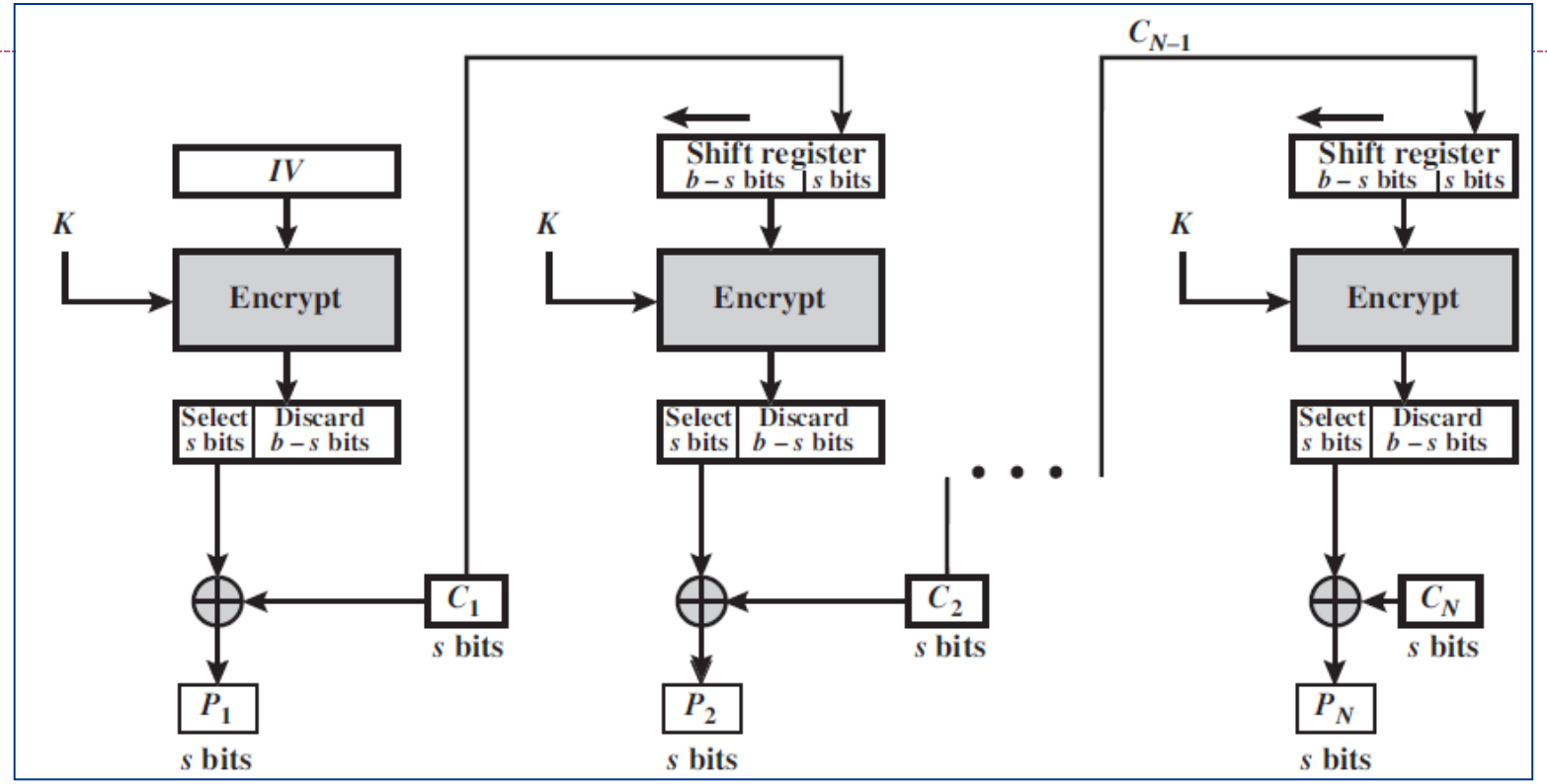
- کاربردهای بی‌درنگ (real time)
- نیاز به پیاده‌سازی رمزگشایی نیست

سبک بازخورد رمز (cipher feedback mode)



- واحد ارسال
  - s بیت
  - معمولا s=8
- متن اصلی به قطعات s بیتی تقسیم می‌شود
- مانند سبک CBC
  - متن رمز شده تابعی از تمامی بیت‌های قبلی در متن اصلی
- در هر مرحله مقدار شیفت رجیستر s بیت شیفت می‌یابد

رمزگشایی CFB



- استفاده از **رمزگذاری** قالبی (نه رمزگشایی)
  - همان الگوریتم رمزگذاری با تنها تفاوت در جهت ورودی‌های XOR

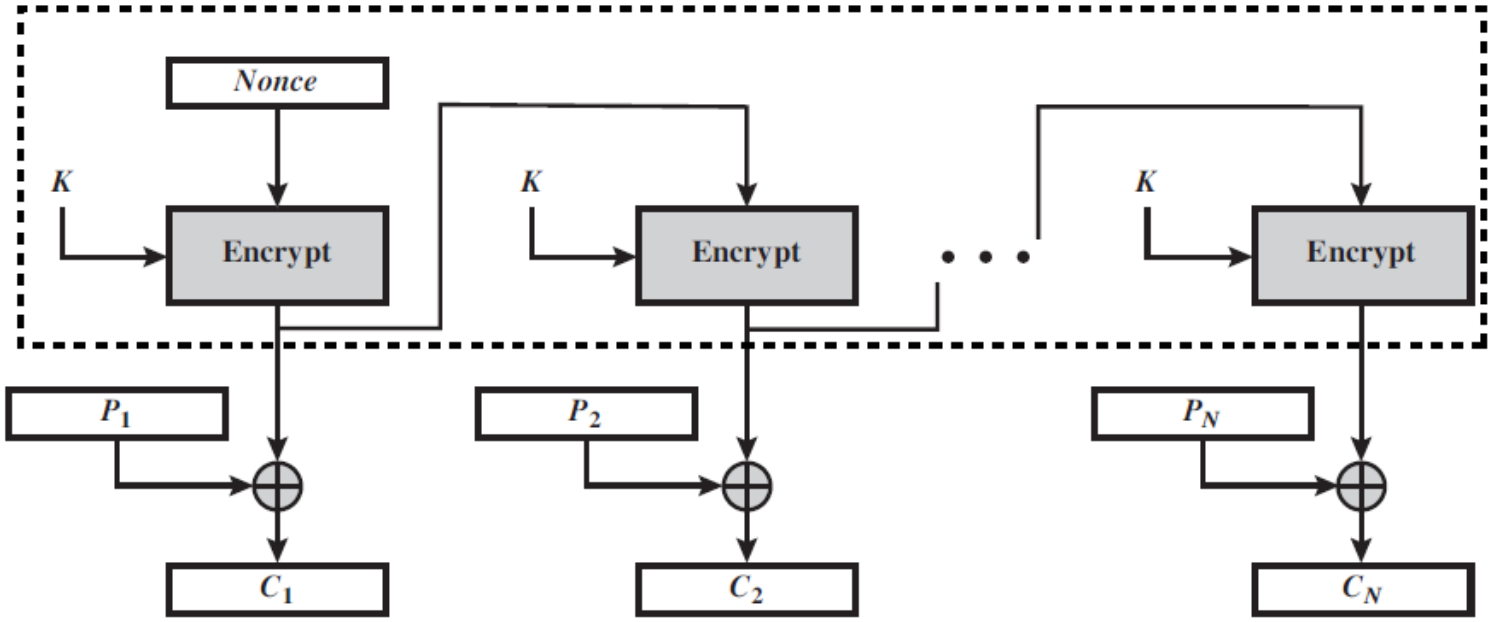
$$C_1 = P_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})] \qquad P_1 = C_1 \oplus \text{MSB}_s[\text{E}(K, \text{IV})]$$

CFB

CFB	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \parallel C_{j-1} \quad j = 2, \ldots, N$ $O_j = \text{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus \text{MSB}_s(O_j) \quad j = 1, \ldots, N$	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \parallel C_{j-1} \quad j = 2, \ldots, N$ $O_j = \text{E}(K, I_j) \quad j = 1, \ldots, N$ $P_j = C_j \oplus \text{MSB}_s(O_j) \quad j = 1, \ldots, N$

- CFB مشابه رمز دنباله‌ای است
- رمز دنباله‌ای: کلید و متن اصلی XOR می‌شوند
- CFB
  - متن اصلی با دنباله‌ای XOR می‌شود که وابسته به متن اصلی است
  - انتشار خطا
  - رمزگذاری قابل موازی‌سازی نیست
  - رمزگشایی قابل موازی‌سازی است

سبک بازخورد خروجی (output feedback mode)
OFB

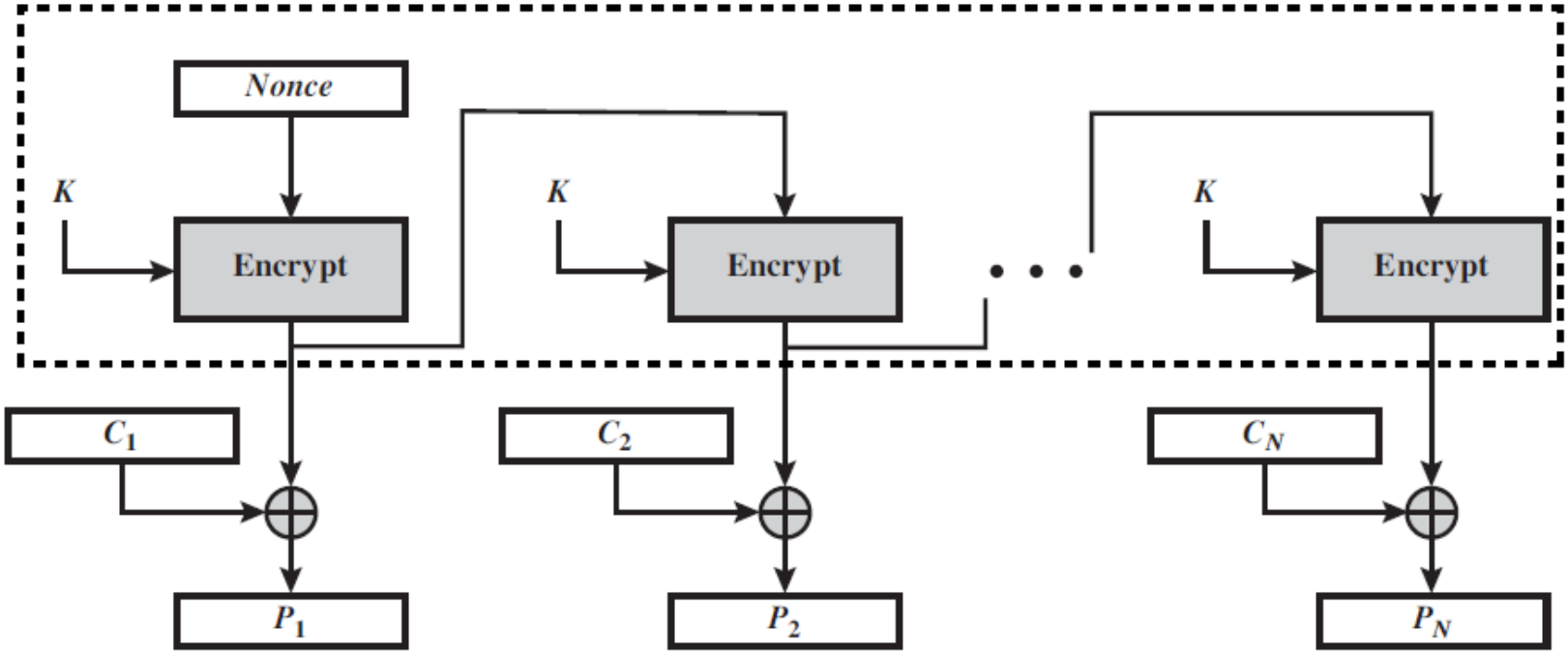


- مشابه CFB
- به جای متن رمزشده، خروجی واحد رمزگذار بازخورد می‌شود
- روی کل قالب متن اصلی عمل می‌کند (حالت s بیتی نیز ممکن است)

$$C_j = P_j \oplus \mathrm{E}(K, [C_{j-i} \oplus P_{j-1}])$$

رمزگشایی OFB

$$P_j = C_j \oplus \mathrm{E}(K, [C_{j-1} \oplus P_{j-1}])$$



OFB

OFB	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus O_j \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = \mathrm{LSB}_{b-s}(I_{j-1}) \parallel C_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $P_j = C_j \oplus O_j \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$

- اگر قالب نهایی کمتر از b بیت باشد، بیت‌های اضافی قالب آخر در متن رمزشده دور ریخته شده و ارسال نمی‌گردد
- IV برای هر بار اجرای الگوریتم باید یکتا باشد
  - خروجی واحد رمزگذار تنها به کلید و IV وابسته است و نه به متن اصلی
- **مشکل انتشار خطا حل شده است**
- عیب OFB: در مقابل حملات تغییر پیام آسیب‌پذیرتر است (امکان کنترل تغییر در متن اصلی پس از رمزگشایی)
  - اگر تغییر یک بیت در متن رمزشده، بیت متناظر در متن اصلی را تغییر دهد

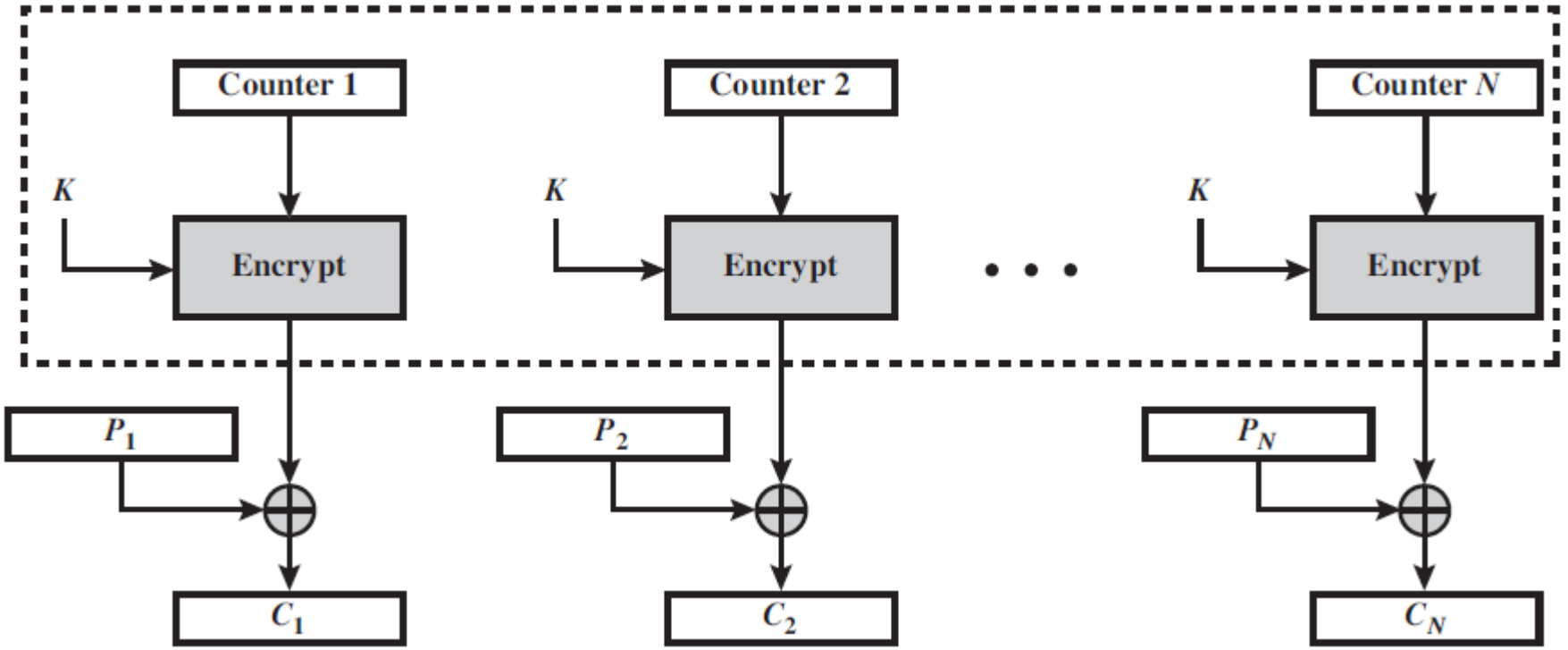
OFB

- شبیه‌تر به رمز دنباله‌ای است
- رمز دنباله‌ای: کلید و متن اصلی XOR می‌شوند
- OFB
  - دنباله کلید بر اساس کلید اصلی و مقدار اولیه تولید می‌شود (مستقل از متن اصلی)
  - متن اصلی با دنباله کلید XOR می‌شود

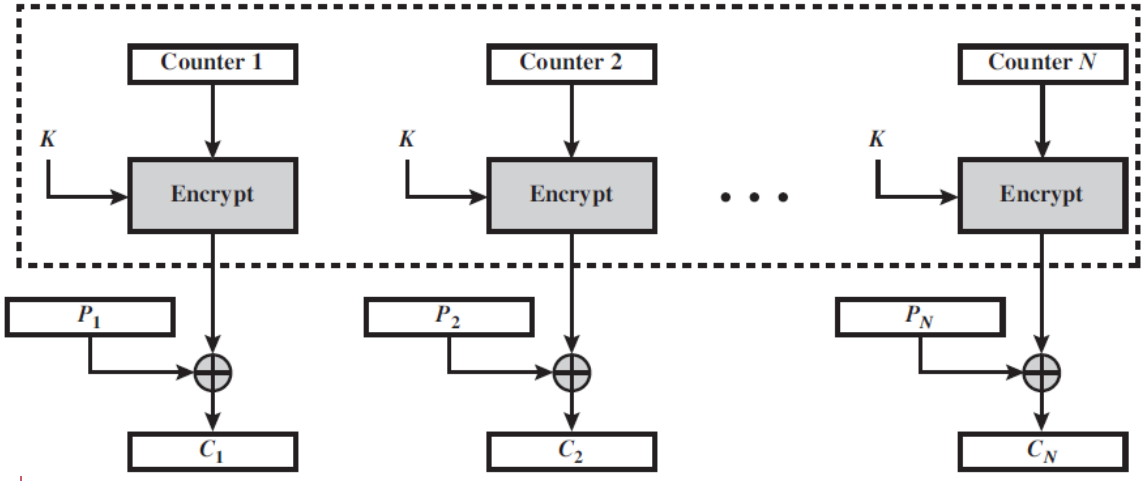
- تفاوت با رمز دنباله‌ای
  - OFB بر قالب‌های کامل متن اصلی (معمولا ۶۴ یا ۱۲۸ بیت) اعمال می‌شود
  - اکثر رمزهای دنباله‌ای به صورت بایتی عمل می‌کنند

سبک شمارنده (counter mode)
CTR

- در دهه ۷۰ پیشنهاد شده ولی اخیرا مورد توجه قرار گرفته است
- ATM (asynchronous transfer mode) network security
- IP sec (IP security)



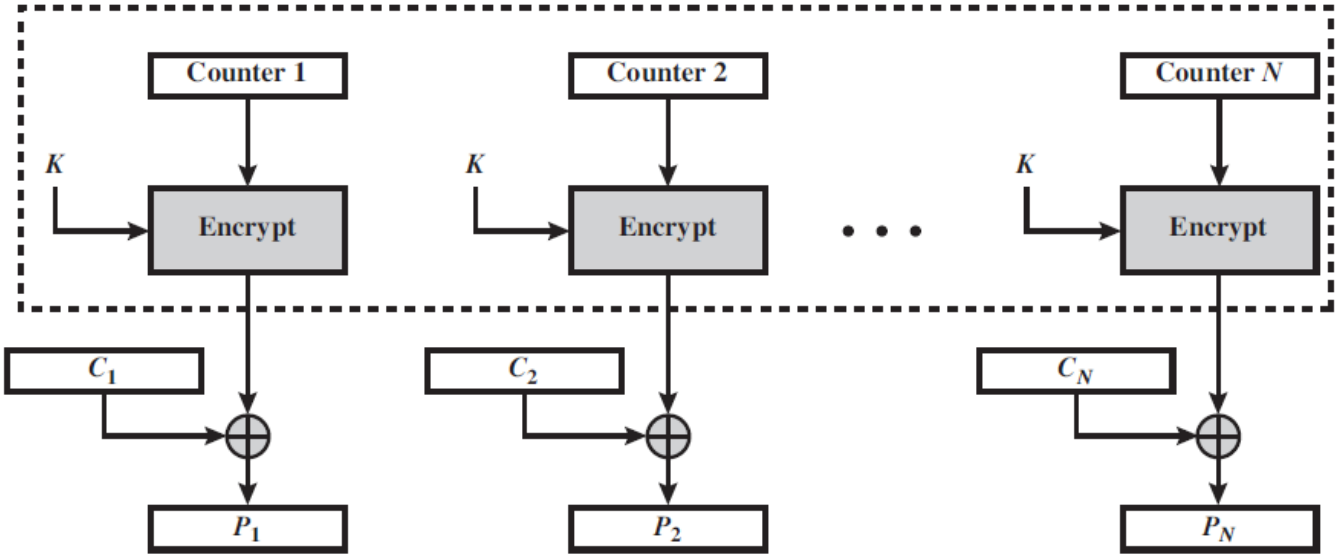
سبک شمارنده (counter mode)
CTR



- یک شمارنده به طول قالب متن اصلی (b بیت)
- SP 800-38A
  - مقدار شمارنده برای هر قالب متن اصلی باید متمایز باشد

- معمولا، مقدار اولیه‌ای برای شمارنده در نظر می‌گیرند و در هر قالب ۱ واحد به آن در پیمانه 2^b اضافه می‌کنند
- زنجیره ندارد
- رمزگشایی: همان الگوریتم با تفاوت در ورودی XOR
  - مقدار اولیه شمارنده مورد نیاز است

CTR



- رمزگشایی

CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$	$P_j = C_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$

CTR

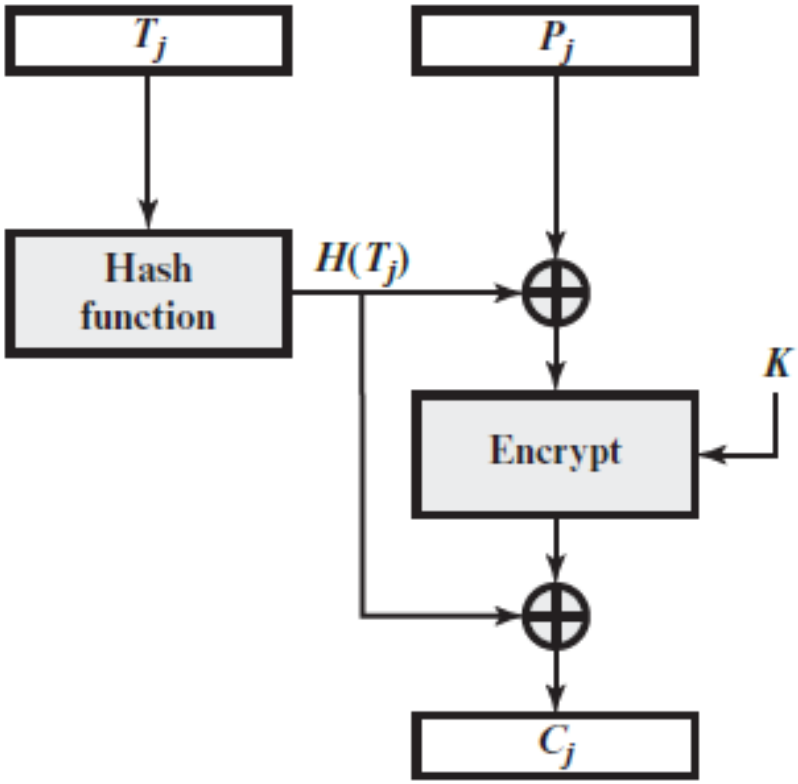
- مقدار اولیه شمارنده (T_1) باید تک‌شمار (nonce) باشد
  - برای تمام پیام‌هایی که با کلید یکسان رمز می‌شوند، مقدار اولیه شمارنده (T_1) باید متمایز باشد
- در طول یک متن اصلی، مقادیر شمارنده (T_i) باید یکتا باشند

مزایا:

- پیاده‌سازی سخت‌افزاری و نرم‌افزاری کارآ
  - رمزگذاری و رمزگشایی قابل موازی سازی است
- پیش‌پردازش: اجرای واحد رمزگذاری مستقل از متن اصلی و متن رمز شده است
  - در صورت وجود حافظه کافی
- دسترسی تصادفی: می‌توان قالب *i*ام ورودی را به طور تصادفی انتخاب و پردازش کرد
- **امنیت قابل اثبات:** CTR حداقل به اندازه سایر سبک‌ها امن است
- **تنها استفاده از واحد رمزگذاری**: مفید در رمزهای قالبی مانند AES

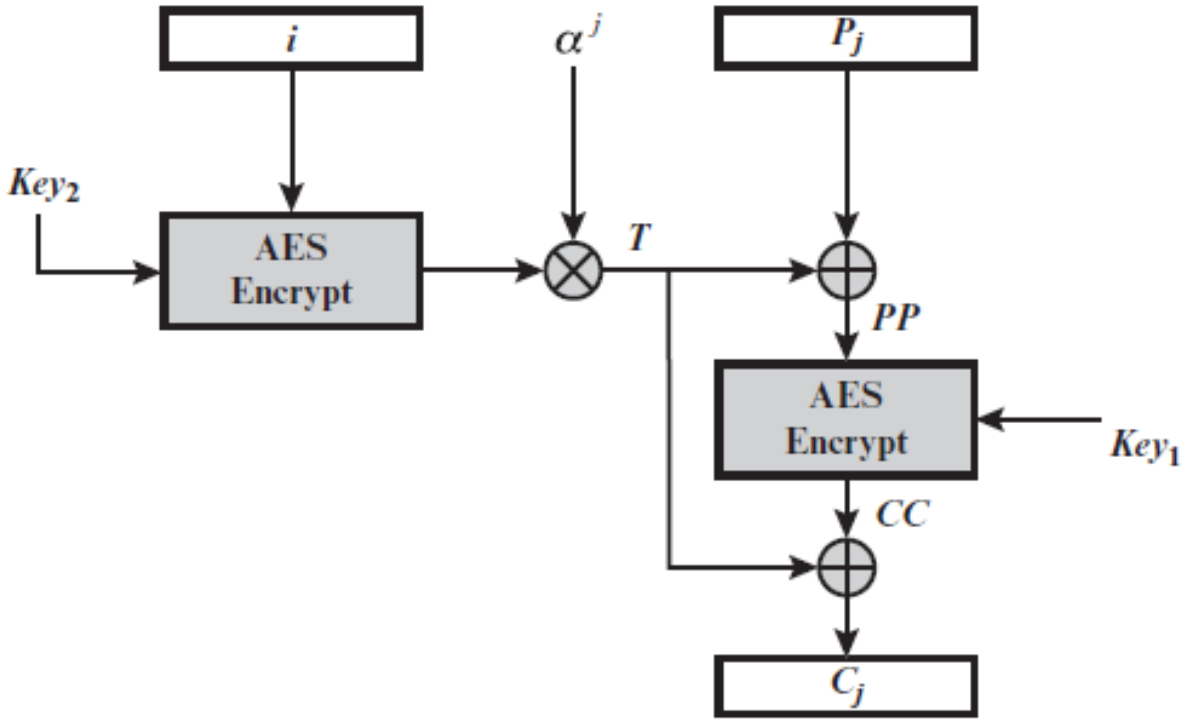
XTS-AES

- تایید شده توسط NIST در ۲۰۱۰ و استاندارد IEEE Std 1619-2007
  - IEEE Security in Storage Working Group (P1619)
  - امنیت داده ذخیره شده در حافظه
  - Tweakable Block Cipher



XTS-AES

- استفاده از سبک ECB با مقدار tweak متفاوت در هر قالب
  - غلبه بر مشکل ECB: اگر قالب b بیتی در متن اصلی بیش از یک بار تکرار شود، متن رمز شده یکسان است



Format-preserving encryption (FPE)

- فرمت متن رمزشده برابر با فرمت متن اصلی باشد

	Credit Card	Tax ID	Bank Account Number
Plaintext	8123 4512 3456 6780	219-09-9999	800N2982K-22
FPE	8123 4521 7292 6780	078-05-1120	709G9242H-35
AES (hex)	af411326466add24 c86abd8aa525db7a	7b9af4f3f218ab25 07c7376869313afa	9720ec7f793096ff d37141242e1c51bd

- کاربرد: امور مالی و اقتصادی، رمزنگاری شفاف و ...
- مزیت:
  - رمزنگاری بخشی از داده
  - استفاده از پروتکل‌ها و نرم‌افزارهای جاری

ساختار فایستلی برای FPE