

# رمزگاری متقاض (مدرس)

Symmetric Encryption

# فهرست مطالب

## ■ رمزهای قالبی و جریانی

- ساختارهای SPN و فایستل و ویژگیهای آشتفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# اعمال مورد استفاده در رمزگذاری مدرن

- در روشهای رمزگذاری مدرن، از آعمال جانشینی و جایگشت استفاده می شود
  - به علاوه، توابعی ساده مانند  $\oplus$  (XOR)
- مجموعه اعمال فوق طی مراحل متوالی روی متن اولیه اعمال می شوند
- تکنیک ماشین های چرخنده الهام بخش روشهای رمزگذاری مدرن بوده است

# الگوریتم های رمز متقارن

■ رمزهای قالبی یا قطعه‌ای (Block Cipher)

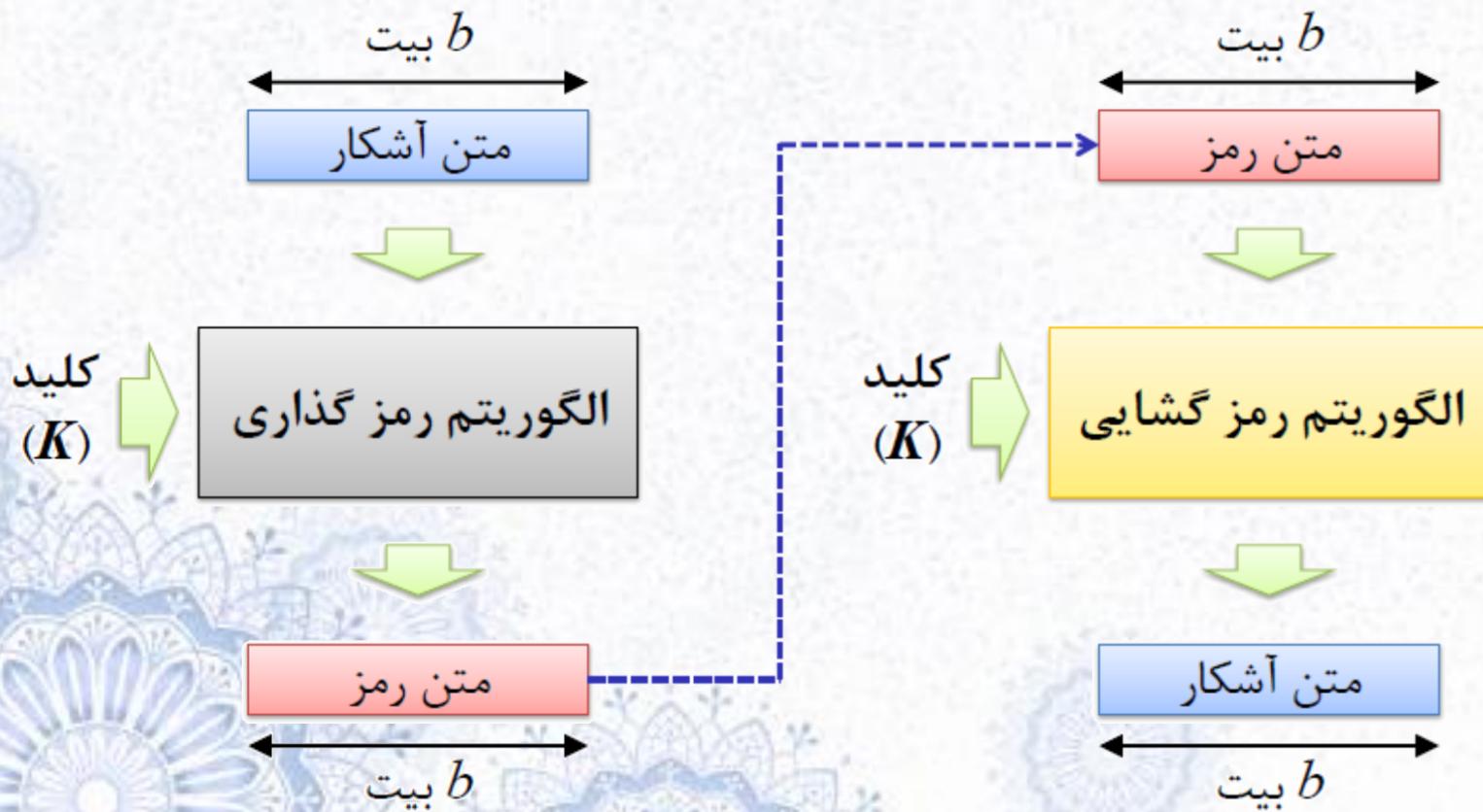
■ پردازش پیغامها به صورت قطعه به قطعه

■ اندازه متعارف مورد استفاده برای قطعات: ۶۴، ۱۲۸ یا ۲۵۶ بیت

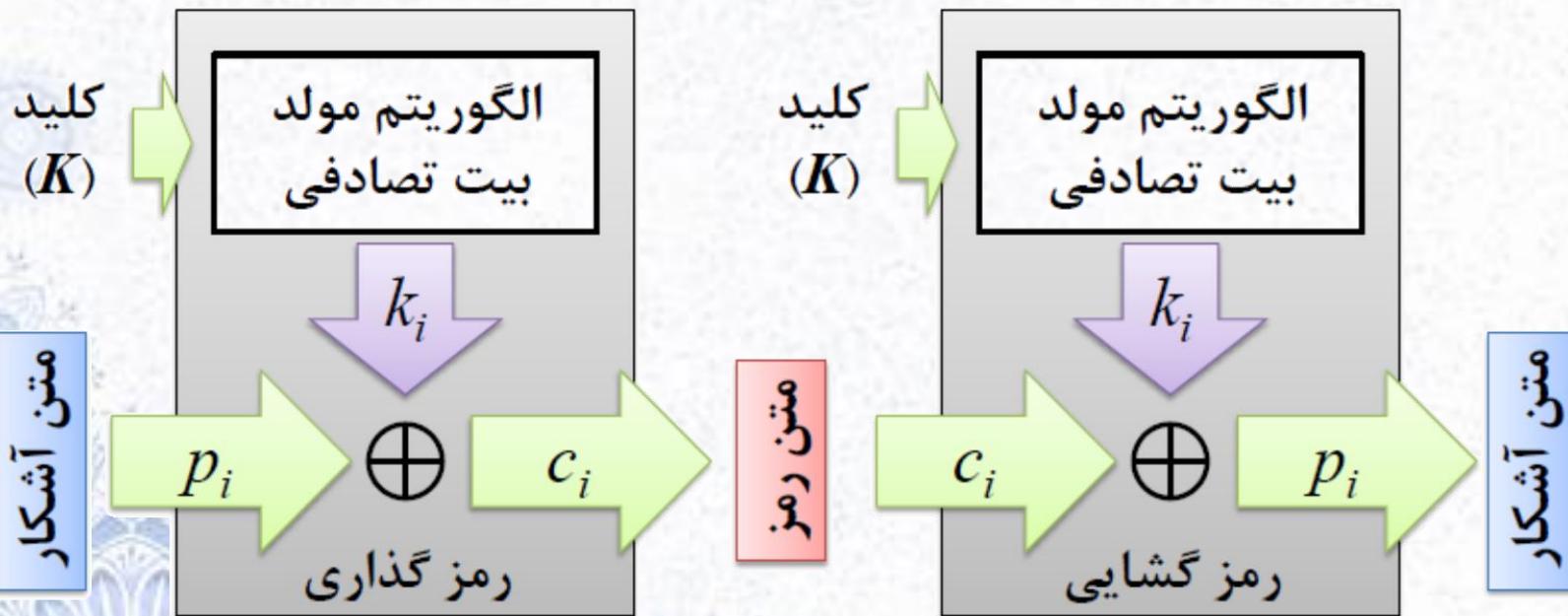
■ رمزهای جریانی (Stream Cipher)

■ پردازش پیغام‌ها به صورت پیوسته (بیت به بیت)

# رمزهای قالبی



# رمزهای جریانی



# فهرست مطالب

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشتفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# ساختار رمزهای قالبی

پیشنهاد شانون: به کارگیری ساختارهای جایگذاری و جایگشت با هدف رسیدن به دو ویژگی:

آشفتگی (Confusion)

پخش (Diffusion)

ایده: تقسیم و حل

تقسیم قطعه و ودی به قسمتهای کوچک

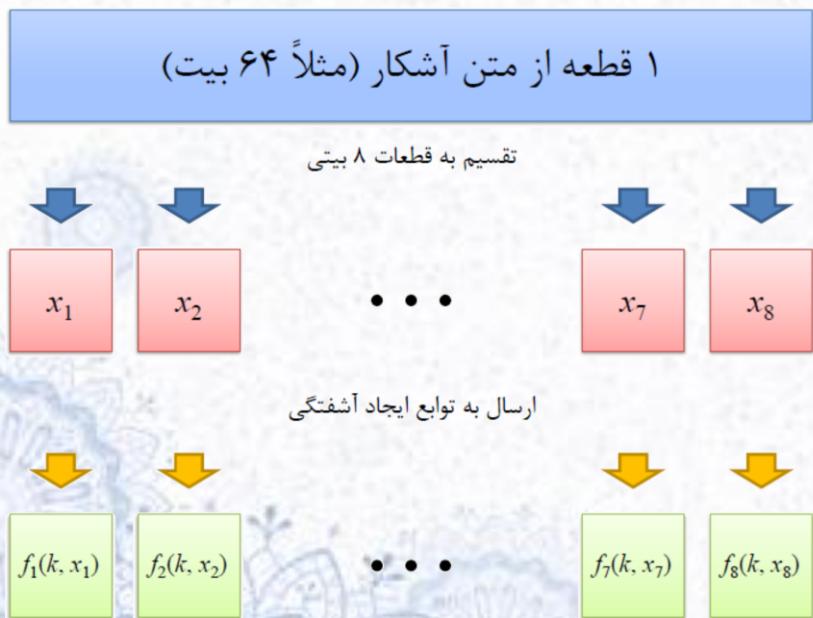
حل مسئله برای هر قسمت

ترکیب راه حلها



Claude E. Shannon  
(1916 – 2001)

# ویژگی آشتفتگی (Confusion)



■ رابطه بین متن رمز و کلید پیچیده باشد  
■ هر بیت از متن (رمز) به بیشتر بیت های کلید وابسته باشد

■ طراحی توابع آشتفتگی با ورودی کوچکتر ساده‌تر است؛ اما تحلیل آن توسط مهاجم نیز آسانتر خواهد بود

■ نیاز به مصالحه  
■ امروزه ورودی ۸ بیتی مناسب به نظر می‌رسد

■ تغییرات توابع آشتفتگی، **موضعی (local)** است.

■ نیاز به پخش تغییرات در متن (رمز) ←  
پخش

# ویژگی پخش (Diffusion) یا پراکندگی

● یعنی ویژگی‌های آماری متن آشکار در ویژگی‌های آماری بخش زیادی از متن رمز پراکنده شود

● هر بیت از متن آشکار در بیت‌های زیادی از متن (همز تاثیر بگذارد)

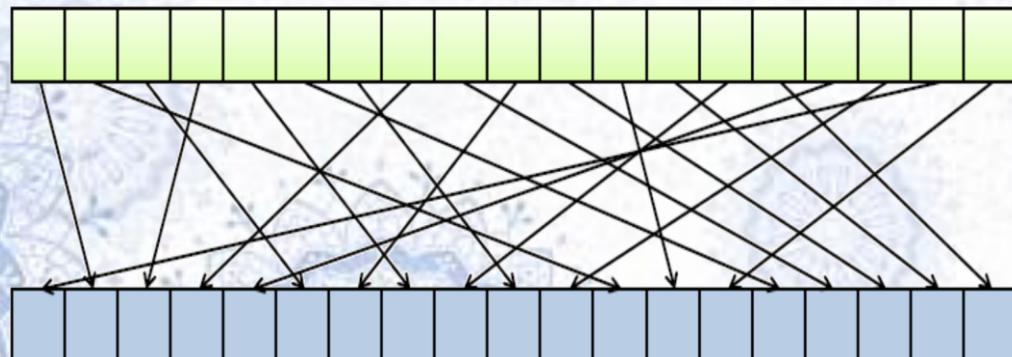
● معمولاً با **جایگشت** بیت‌های ورودی پیاده‌سازی می‌شود

● پگونگی جایگشت می‌تواند وابسته به **کلید** باشد

● ولی در کل جایگشت با پخش معادل نیست (اشتباه (ایچ))

● ترکیب جایشگت با اعمال دیگر در دورهای مختلف باعث **پخش** می‌شود

● جایگشت اثر موضعی توابع آشفتگی (ا پخش می‌کند



# دور

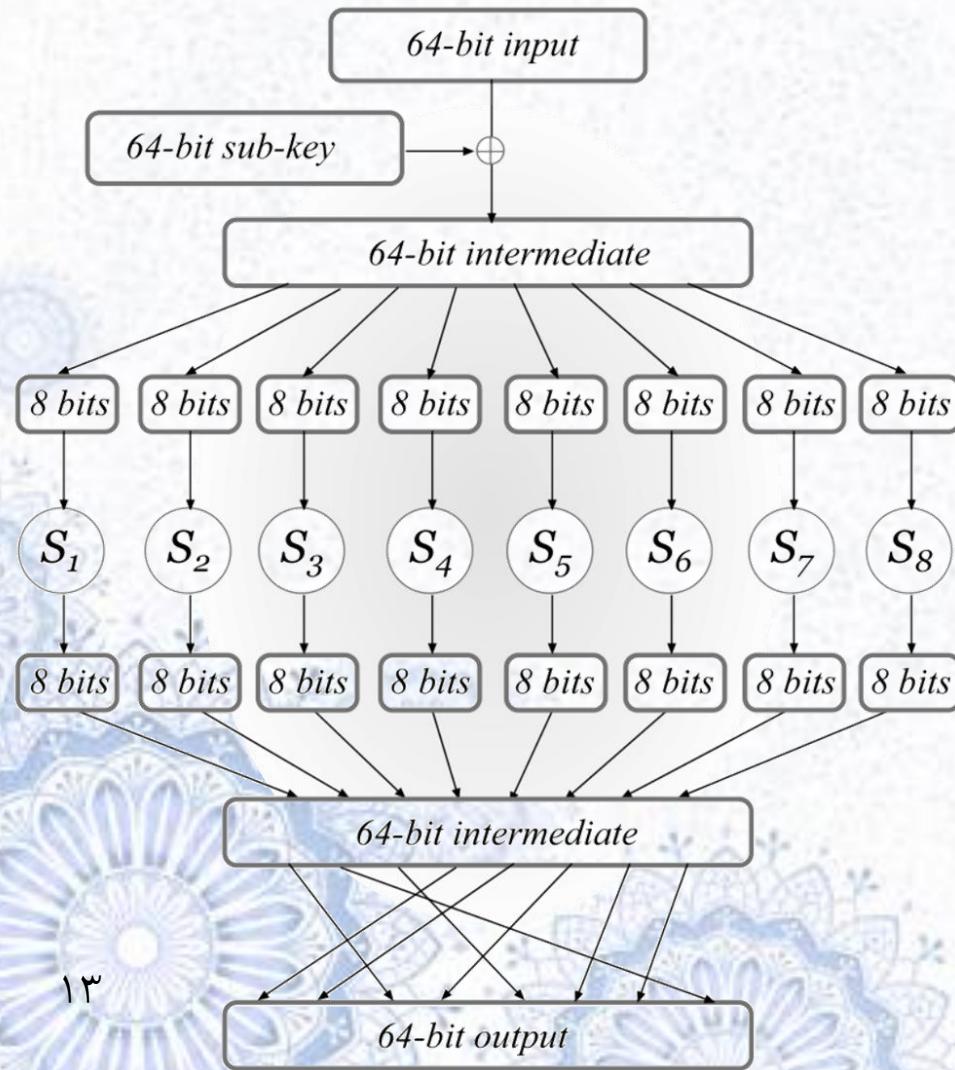
- ❖ دور (round): یک مرحله جابجایی + یک مرحله جایگشت
- ❖ یک الگوریتم رمز از چندین دور تشکیل می شود
- ❖ **جایجایی و جایگشت هر دور می تواند متفاوت باشد**
- ❖ با افزایش تعداد دور، رمز پیچیده تر شده ولی کارایی آن کمتر می شود
- ❖ یکی از وظایف طراح رمز آن است که تعداد دور بهینه را پیدا کند

# شبکه جانشینی - جایگشتی (SPN)

## /Substitution-Permutation Network

- 🔒 نوع خاصی از الگوی آشفتگی-پخش
- 🔒 توابع دور آن، شکل مشخصی دارند
  - 🔓 غیر وابسته به کلید، آشکار برای همگان (متی مهاجم)
- 🔒 تابع جانشینی: **S**-box یا **S**
- 🔒 تابع جایگشت: **P**-box یا **P**
- 🔒 کلید در ابتدای هر دور با مقدار ورودی XOR می‌شود
  - 🔓 کلید به طور مستقیم (روی اعمال آشفتگی و پخش) تاثیر ندارد

# یک دور از SPN



۱. کلید XOR

۲. اعمال S-BoX

۳. اعمال P-BoX

# مزایا و معایب SPN

▪ مزیت: امکان موازی سازی و در نتیجه افزایش کارایی  
▪ در هر دور می توان S-Boxها را به طور موازی اجرا نمود

▪ عیب ۱: محدودیت طراح در انتخاب S-Box  
▪ S-Boxها باید برگشت پذیر باشد تا بتوان (مزگشایی) نمود  
▪ دست طراح در انتخاب S-Box بسته است

▪ عیب ۲: الگوریتم رمزگشایی با رمزگذاری متفاوت است  
▪ افزایش حجم پیاده سازی، به ویژه در سخت افزار

# شبکه فایستل

ایده: ایجاد رمزی که در آن S-Box‌ها می‌توانند برگشت ناپذیر باشند  
هورست فایستل (Horst Feistel)

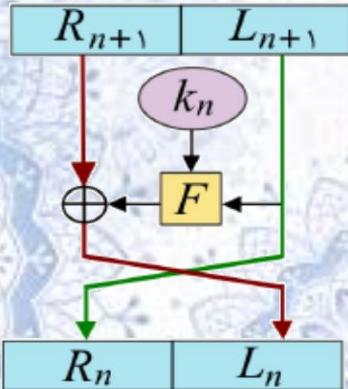
(مزنگار آلمانی شرکت IBM)

یک دور از شبکه فایستل:

رمزگشایی:

$$\begin{aligned} R_n &\leftarrow L_{n+1} \\ L_n &\leftarrow F(L_{n+1}, k_n) \oplus R_{n+1} \end{aligned}$$

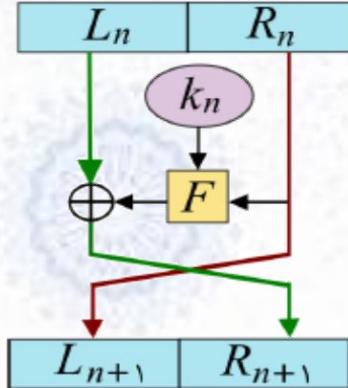
رمزگشایی



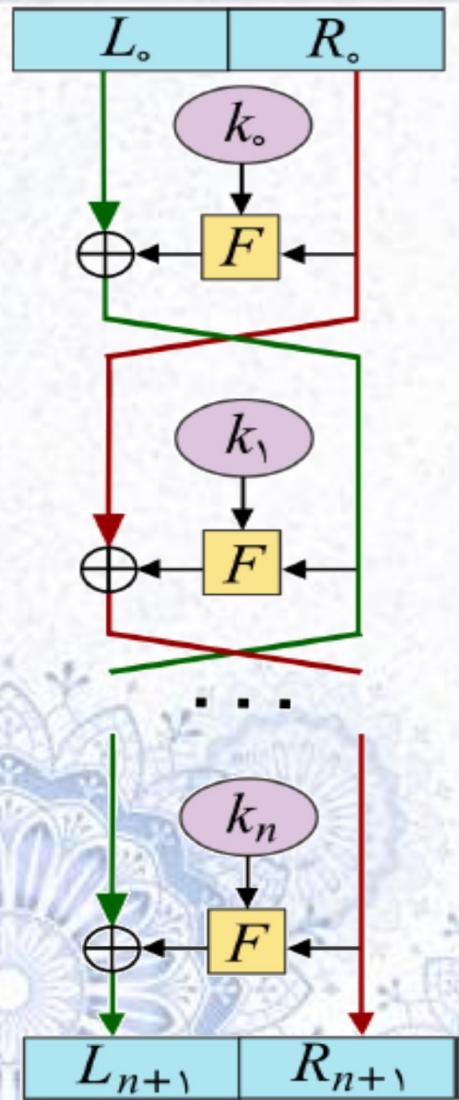
رمزگذاری:

$$\begin{aligned} L_{n+1} &\leftarrow R_n \\ R_{n+1} &\leftarrow F(R_n, k_n) \oplus L_n \end{aligned}$$

رمزگذاری



# ساختار کامل شبکه فایستل



رمزگذاری

$F$  : تابع دور (در تمام دورها ثابت است)

$k_n$  تا  $k_0$ : زیر کلیدها

زیر کلیدها از کلید اصلی مشتق می‌شوند



# پارامترهای شبکه فایستل

- اندازه قطعه
- طول کلید
- تعداد دورها
- الگوریتم تولید زیر کلیدها (کلیدهای دور)
  - تابع دور ( $F$ )
  - لزومی ندارد برگشت پذیر باشد
  - در واقع اگر برگشت ناپذیر باشد امنیت رمز بیشتر می شود

# فهرست مطالب

---

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشتفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# تاریخچه DES

## 锁 طراحی توسط IBM (۱۹۷۴)

锁 به درخواست مؤسسه استاندارد آمریکا (NBS)

锁 برای (مزنگاری اطلاعات غیر طبقه بندی شده ولی مساقط

锁 مبتنی بر الگوریتم لوسيفر (Lucifer)

锁 ابداع شده توسط فایستل

锁 طول کلید کمتر نسبت به لوسيفر

锁 تغیر S-box ها بدون ارائه دلیل

lock احتمال دفاتر آژانس امنیت ملی (NSA)

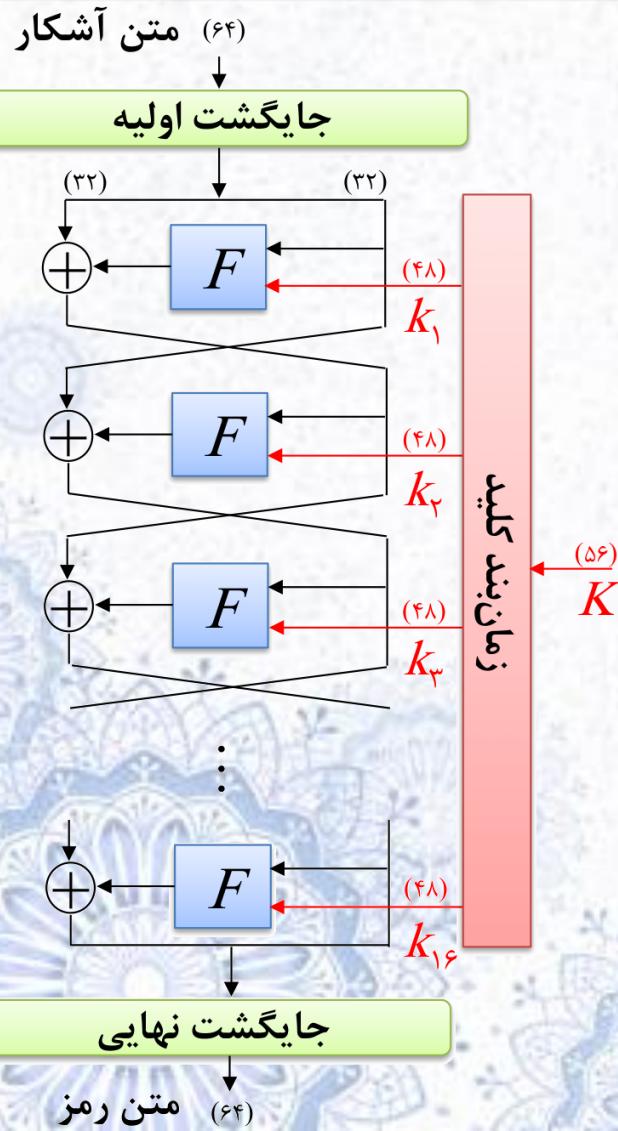
lock بدگمانی به وجود آمده: شاید NSA به گونه‌ای الگوریتم را تغییر داده که فقط خود بتواند آن را (مزکشایی کند

lock در سال ۱۹۹۰ مشخص شد

lock تغییرات اعمال شده NSA به منظور مقاومت در برابر روش تحلیل تفاضلی بوده است

lock فقط NSA و IBM از روش تحلیل تفاضلی مطلع بودند ۹۱۹۱۹۱۹۱۹۱۹۱

# مشخصات عمومی DES



🔒 طول کلید: ۵۶ بیت

🔒 طول قطعه: ۶۴ بیت

🔒 تعداد دورها: ۱۶ دور

🔒 آزمون جامع روی کلید DES حدود ۲ سال زمان نیاز دارد

🔒 با فرض امتحان هر کلید در یک نانو ثانیه  
امکان کاهش پیشنهادی با اجرای موازی

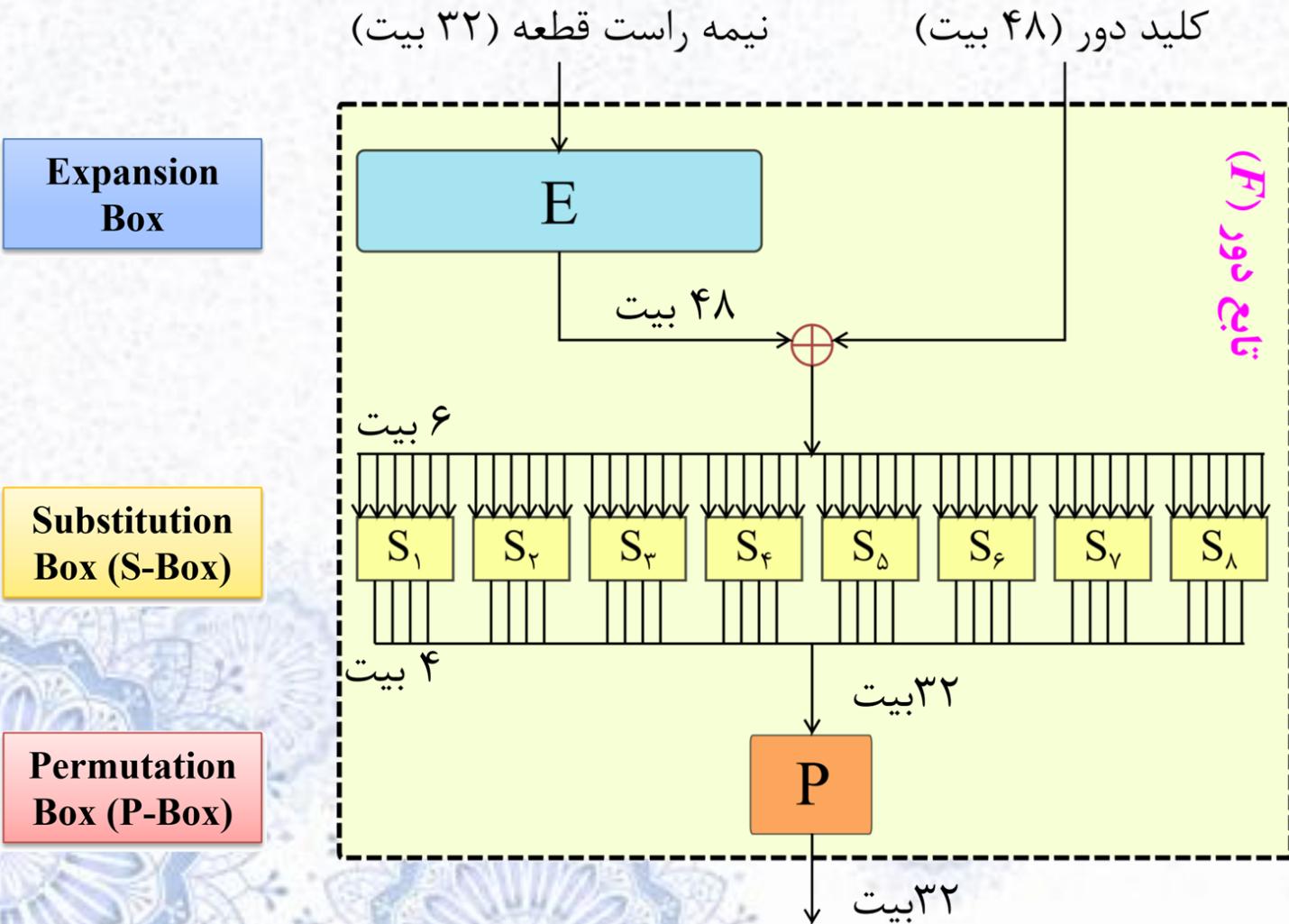
🔒 جایگشت اولیه و نهایی

🔒 عکس هم هستند

🔒 تاثیری در امنیت DES ندارند

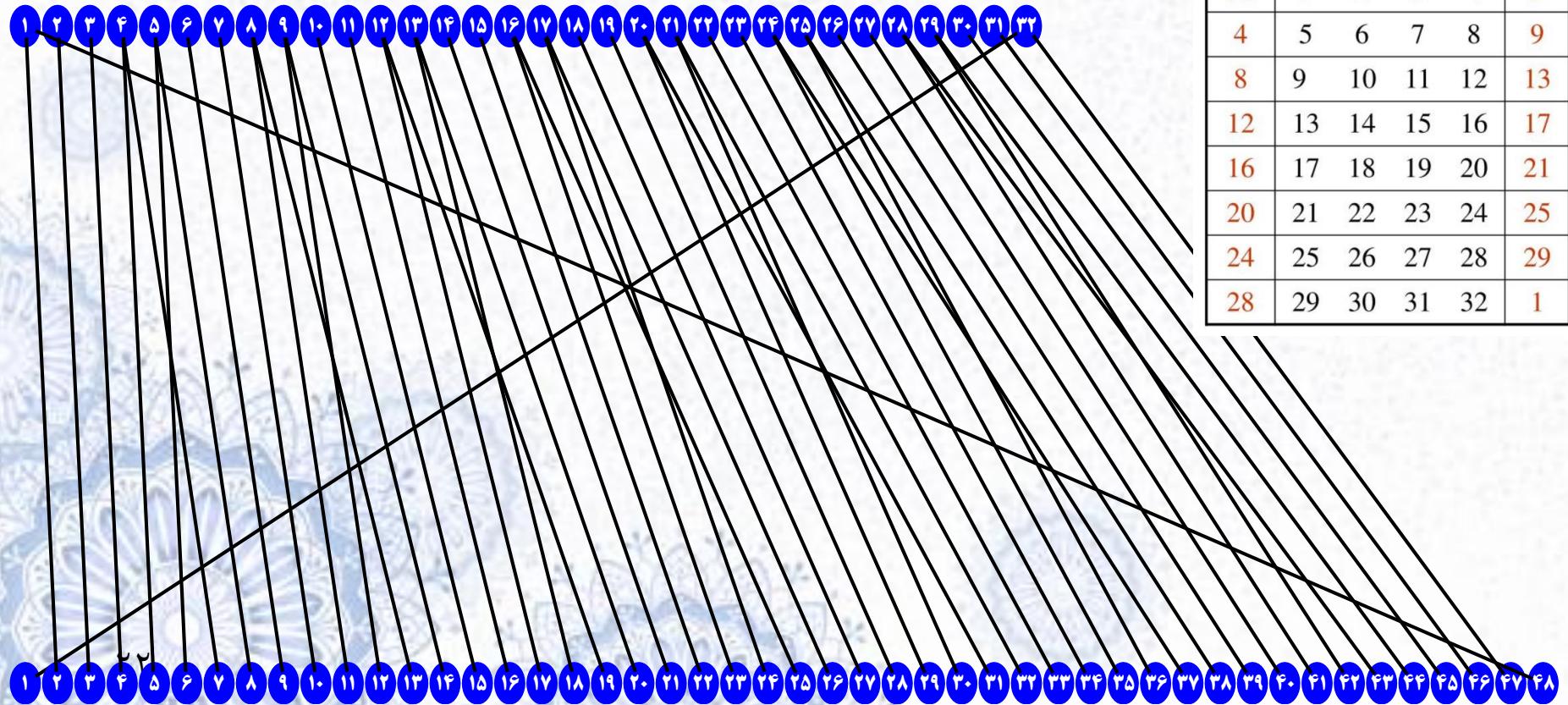
هدف: تسهیل پیاده‌سازی سفت افزاری

# ساختار داخلی تابع دور (F)



# جعبه توسعه (Expansion Box)

تبدیل ۳۲ بیت به ۴۸ بیت با تکرار برخی از بیت‌ها



# ساختار S-Box های DES

**جدول زیر: نخستین S-Box در DES (یعنی ۱)**

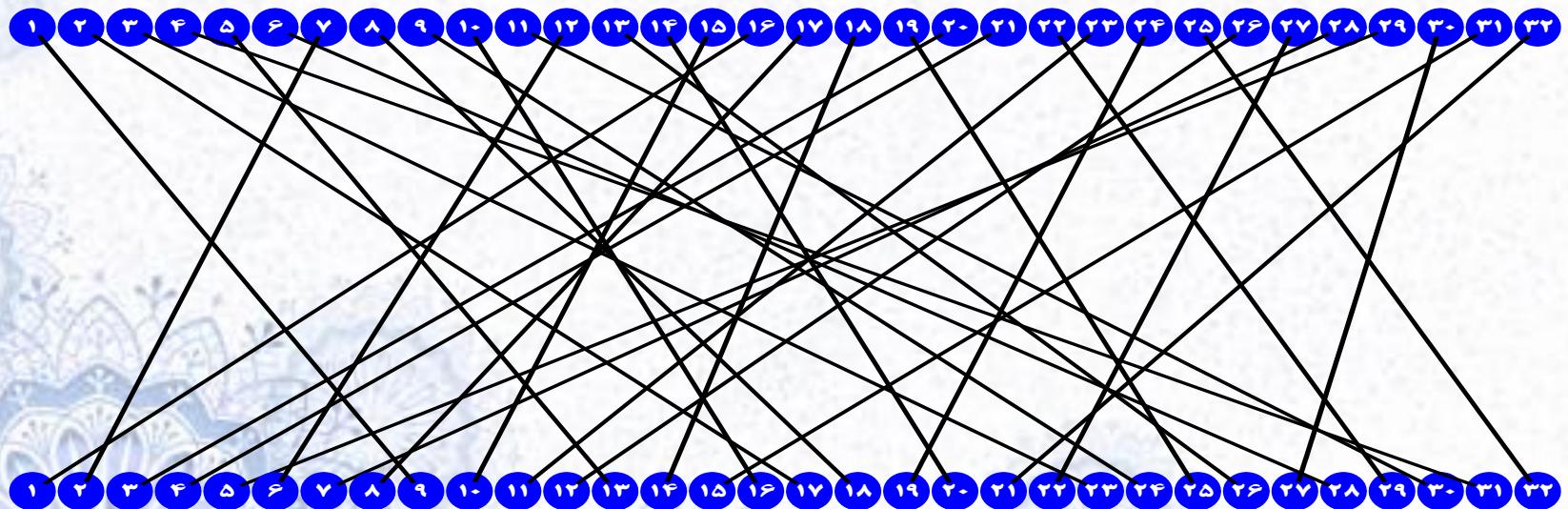
**مثال: ۹ (۰۹۰۱) فروجی: ۰ (۱۱۰۰)**

شماره ستون																شماره سطر ↓
≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	≡	
۷	۰	۹	۵	۱۲	۶	۱۰	۳	۸	۱۱	۱۵	۲	۱	۱۳	۴	۱۴	۰۰
۸	۳	۵	۹	۱۱	۱۲	۶	۱۰	۱	۱۳	۲	۱۴	۴	۷	۱۵	۰	۰۱
۰	۵	۱۰	۳	۷	۹	۱۲	۱۵	۱۱	۲	۶	۱۳	۸	۱۴	۱	۴	۱۰
۱۳	۶	۰	۱۰	۱۴	۳	۱۱	۵	۷	۱	۹	۴	۲	۸	۱۲	۱۵	۱۱

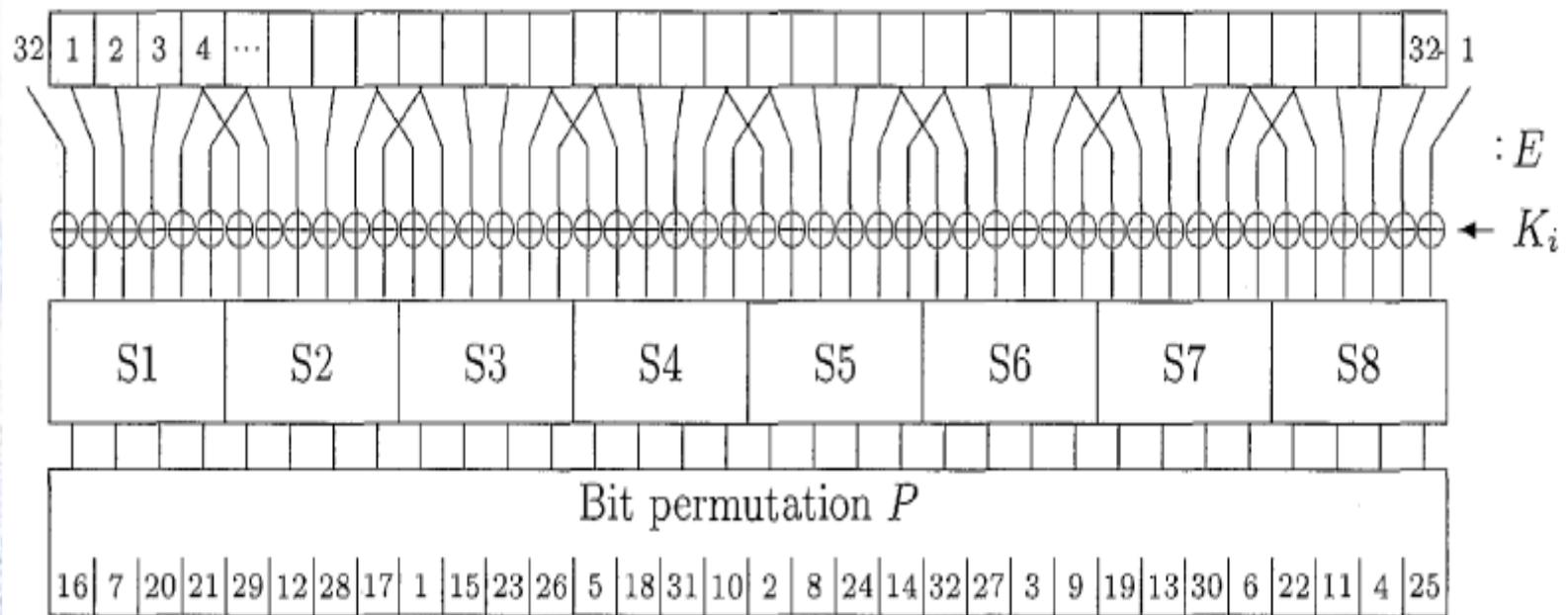
**هشت S-Box در مجموع ۴۸ بیت ورودی را به ۳۲ بیت تبدیل می کنند**

# P-Box

آخرين گام از تابع F در DES است.  
P-Box ۳۲ بيت ورودي به ۳۲ بيت خروجي جايگشت داده می شوند.



# نگاهی کلی به تابع دور DES



# تابع بولی خطی

• Y را تابع خطی از بیتهاي  $x_1, \dots, x_n$  می نامیم اگر بیتهاي چون  $a_0 \dots a_1$  وجود داشته باشد به قسمی که:

$$Y(x_1, \dots, x_n) = a_0 \oplus (a_1 \cdot x_1) \oplus \dots \oplus (a_n \cdot x_n)$$

• نقطه (.) نماد عمل AND است  
• به سادگی می توان تحقیق کرد که تنها بخشی از تابع F که تابعی خطی از ورودی نیست، S-Box است

• اگر S-Box ها نبودند، متن رمز تابعی خطی از متن آشکار شده و شکستن DES بسیار آسان می شد

# زمان بندی کلید

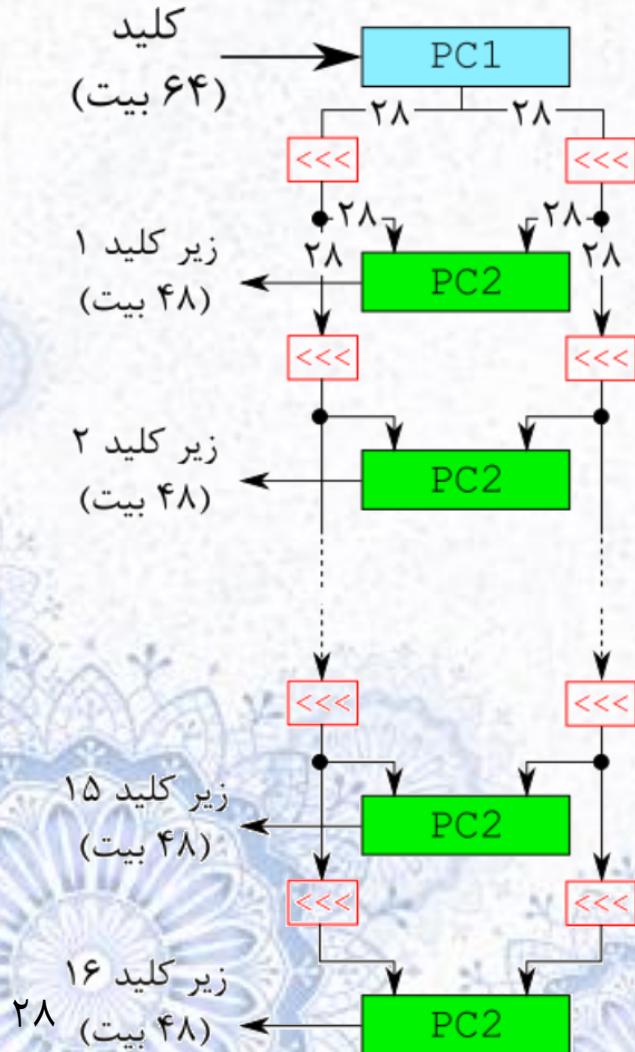
☞ هر بیت کلید حدوداً در ۱۴ دور از ۱۶ دور استفاده می‌شود.

$$\frac{48 * 16}{56} \approx 13.7$$

☞ زمان بندی کلید، یک مقدار ۶۴ بیتی را به عنوان کلید می‌پذیرد  
☞ طول کلید DES [بایت ۸ بایت باشد](#)

☞ ولی فقط ۵۶ بیت مشخص از آن استفاده می‌شود  
☞ بقیه بیتها به عنوان parity کلید مورد استفاده قرار می‌گیرند  
☞ بیت هشتم از هر بایت، parity آن بایت است

# ساختار درونی زمان بند کلید



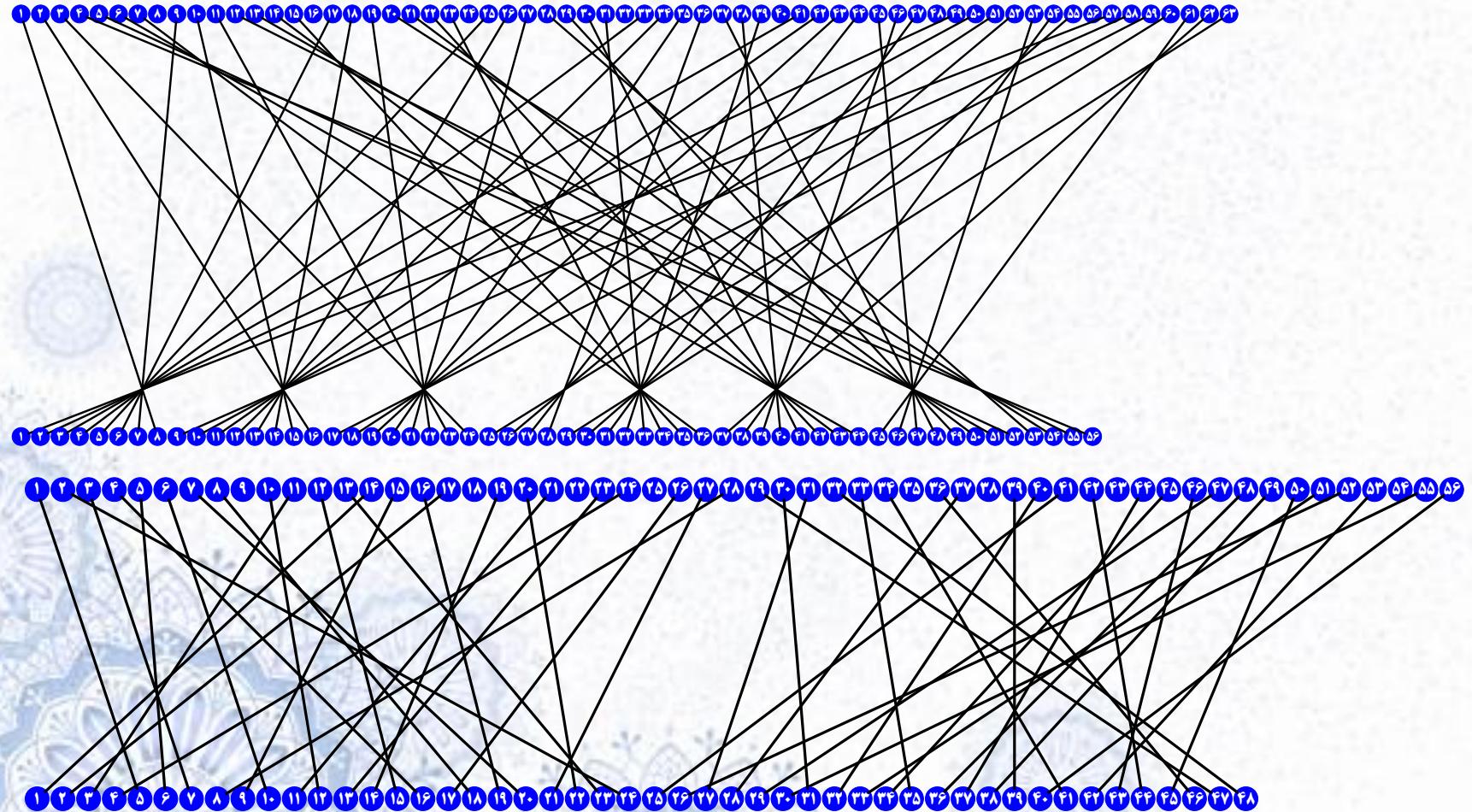
## PC: Permuted Choice

هدف: جایگشت دادن و انتخاب برخی بیتها

PC1: جایگشت و انتخاب ۵۶ بیت از ۶۴ بیت ورودی  
PC2: جایگشت و انتخاب ۴۸ بیت از ۵۶ بیت ورودی

- Rotate <<<
- نماد چرخش به چپ (Left) بیتها
  - در دورهای ۱، ۲، ۹ و ۱۶ فقط ۱ بیت چرخش
  - در سایر دورها ۲ بیت چرخش

# ساختار PC1 و PC2



# اثر بهمنی (Avalanche Effect)

اثر بهمنی: با تغییر کوچک در بیتهاي کلید يا متن آشکار، تغییر زیادی در بیتهاي متن رمز رخ دهد

اثر بهمنی اكيد (SAC) يا Strict Avalanche Criterion

صوری سازی (Formalization) مفهوم فوق

با عوض کردن هر بیت در ۹۰ درصد، هر یک از بیتهاي خروجي با احتمال حدوداً ۵۰٪ عوض شوند

# چرا بی تعداد دور DES

چرا تعداد دور DES برابر ۱۶ است؟

- بعد از ۵ دور، هر بیت از متن (همز تابعی از تماه بیتها متن و آشکار و تماه بیتها) کلید است
- بعد از ۸ دور، خاصیت SAC برقرار می شود

در حمله **تفاضلی** ثابت می شود که اگر تعداد دور کمتر یا مساوی ۱۵ باشد:

- می توان DES را با یک حمله متن آشکار معلوم (KPA) سریعتر از حمله جامع شکست
- طراهمان DES سالها قبل از کشف حمله تفاضلی با آن آشنا بودند

# فهرست مطالب

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشتفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- **الگوریتمهای رمز 3DES و 2DES**
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# و حمله آزمون جامع DES

DES در مقابل حمله آزمون جامع آسیب‌پذیر است

به فاطر طول کلید کوتاه

راه حل:

استفاده از الگوریتم های (مزنگاری دیگر)

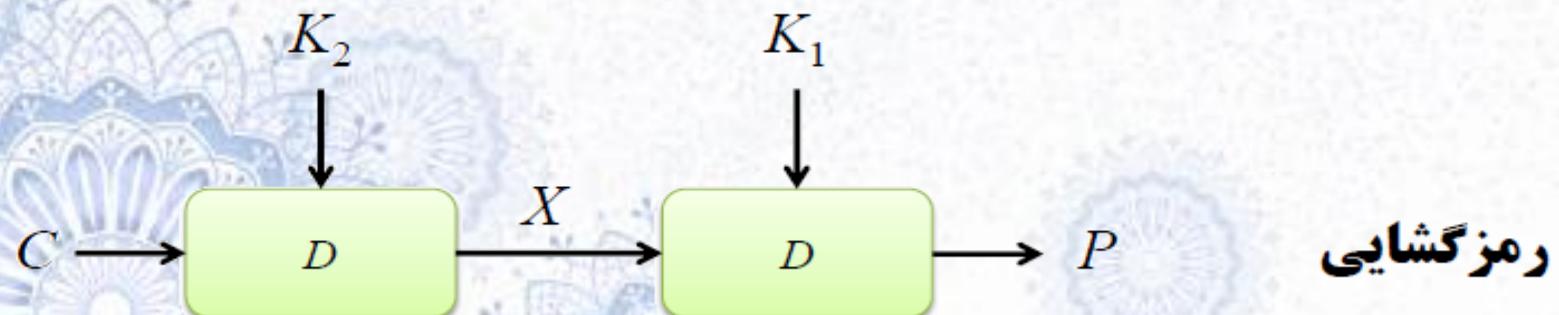
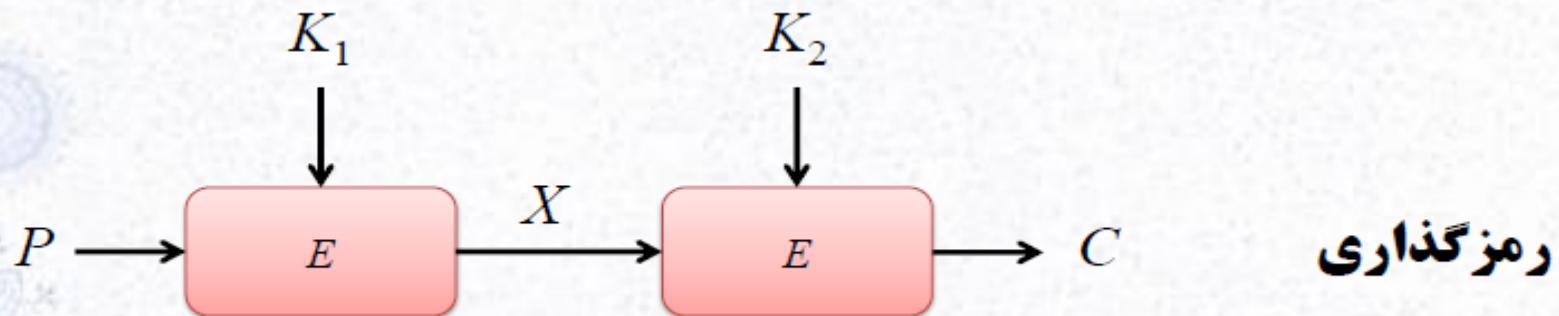
پیمیده کردن الگوریتم DES از طریق اضافه کردن مراحل (مزنگاری و افزایش طول کلید)

چند مرتبه تکرار DES با کلیدهای مستقل

# الگوریتم 2DES (دابل - دز)

دو مرتبه رمزگاری با دو کلید مستقل

طول کلید = ۱۱۲ بیت



# حمله ملاقات در میانه (Meet-in-the-Middle)

با داشتن یک زوج کلید متن آشکار و متن رمز (P,C) :

$$C = E(K_2, E(K_1, P)) \quad E(K_1, P) = X = D(K_2, C)$$

روش حمله

P را با همه  $2^{56}$  کلید ممکن برای  $K_1$  رمز و نتایج و ذخیره کن

C را با همه  $2^{56}$  کلید ممکن برای  $K_2$  (مزگشایی مقادیر حاصل را با نتایج ذخیره شده مقایسه کن

در صورت تطابق زوچ کلید را یافته اید

معادل آزمون جامع برای  $2^{57}$  حالت

کاهش زمان جستجو با استفاده از هافظه بیشتر

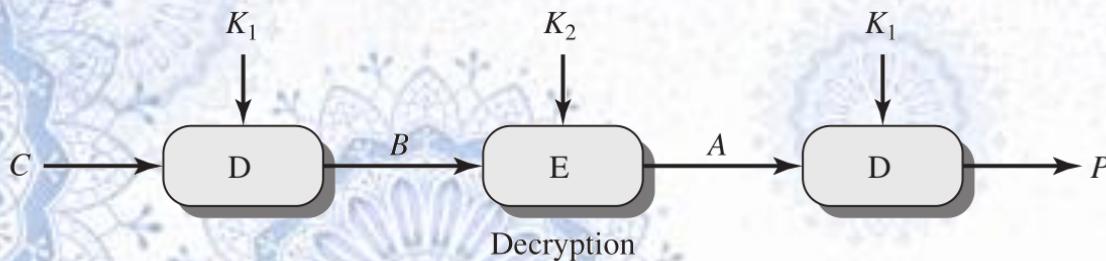
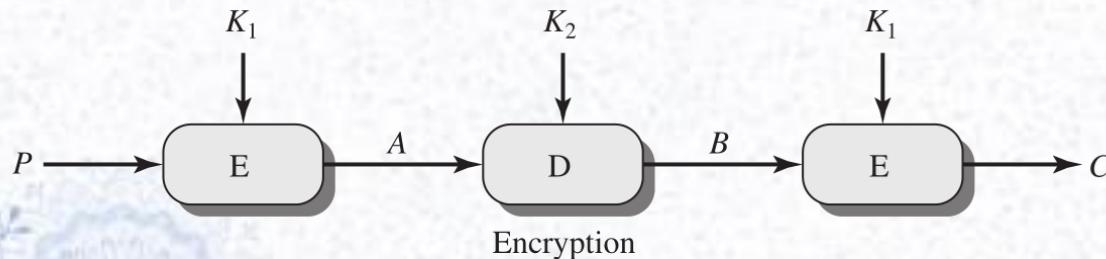
بنابراین 2DES بهبود محسوسی نسبت به DES حاصل نمی کند

# الگوریتم 3DES (تریپل دز) با دو کلید

حل مشکل 2DES با سه مرحله رمزگذاری با DES  
امکان بهره‌گیری از ۳ بار DES با دو کلید به صورت یر:

$$C = E(K_1, D(K_2, E(K_1, P)))$$

$$P = D(K_1, E(K_2, D(K_1, C)))$$



# الگوریتم 3DES با سه کلید

استفاده از سه کلید مختلف 

$$C = E(K_3, D(K_2, E(K_1, P)))$$

کلید = ۱۶۸ بیت 

تا کنون روی 3DES دو یا سه کلیدی حمله عملی گزارش نشده است  
 ولی بسیار کند است!

استفاده از ترتیب EDE برای رمزگذاری: 

با مساوی قرار دادن هر سه کلید به یک DES معمولی هی (سیم)  
 سازگاری با تجهیزاتی که 3DES را نمی‌فهمند

# فهرست مطالب

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# استاندارد رمزگذاری پیشرفته AES

مسابقه‌ی NIST ( مؤسسه ملی علم و فناوری آمریکا ) در سال ۱۹۹۷ برای طراحی الگوریتم رمز

برنده: رمز ریندا (Rijndael) در سال ۲۰۰۰

سایر نامزدهای فینال:

IBM از MARS ▪

RSA از RC6 ▪

Serpent ▪

Twofish ▪

پشتیبانی از طول کلید ۱۲۸، ۱۹۲ و ۲۵۶

طول کلید استاندارد: ۱۲۸

مبتنی بر ساختار رمز SPN است ولی از نوع فایستل نیست

کلید ۱۲۸ بیتی ( ۴ کلمه‌ای )، به یک آرایه W با ۴۴ کلمه‌ی ۳۲ بیتی بسط داده می‌شود

کلید هر دور ۴ عنصر این آرایه ( ۱۲۸ بیت ) است

# نحوه کار AES-128

الگوریتم زمانبندی کلید نقش تهیه کلید برای هر دور بر اساس کلید اصلی را بر عهده دارد.

برخلاف DES و بسیاری از رمزهای دیگر، اعمال لازم بر روی **بایتها** انجام می‌شود، نه بیتها

متون آشکار ۱۲۸ بیتی به شکل یک ماتریس حالت  $4 \times 4$  در می‌آید

هر درایه یک بایت از متن آشکار را نشان می‌دهد

این ماتریس به صورت **ستونی** پر می‌شود

این ماتریس در انتهای شامل متن رمز خواهد بود

# نحوه کار AES-128

متن آشکار ورودی به صورت ستونی در ماتریس حالت ذخیره می شود.

Input = 32 43 f6 a8 88 5a 30 8d  
31 31 98 a2 e0 37 07 34



32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

# مراحل رمزگذاری AES-128

■ در هر دور ۴ عمل برگشت‌پذیر روی ماتریس حالت اعمال می‌شود  
■ جانشینی با آنها:

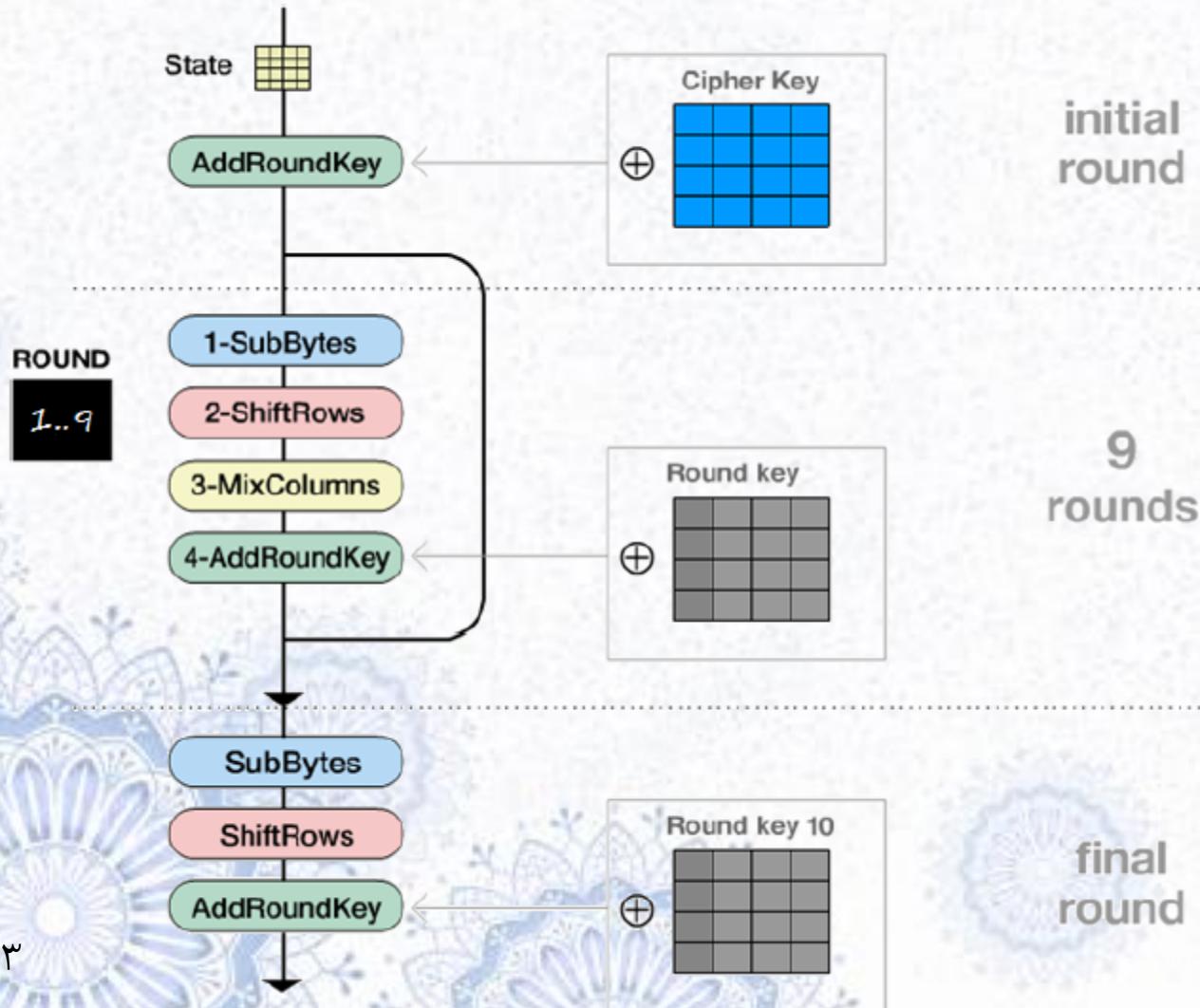
■ جانشینی درایه‌های ماتریس حالت با استفاده از یک S-Box  
■ شیفت سطري:

■ شیفت دورانی سطرهای ماتریس حالت  
■ ترکیب ستونها:

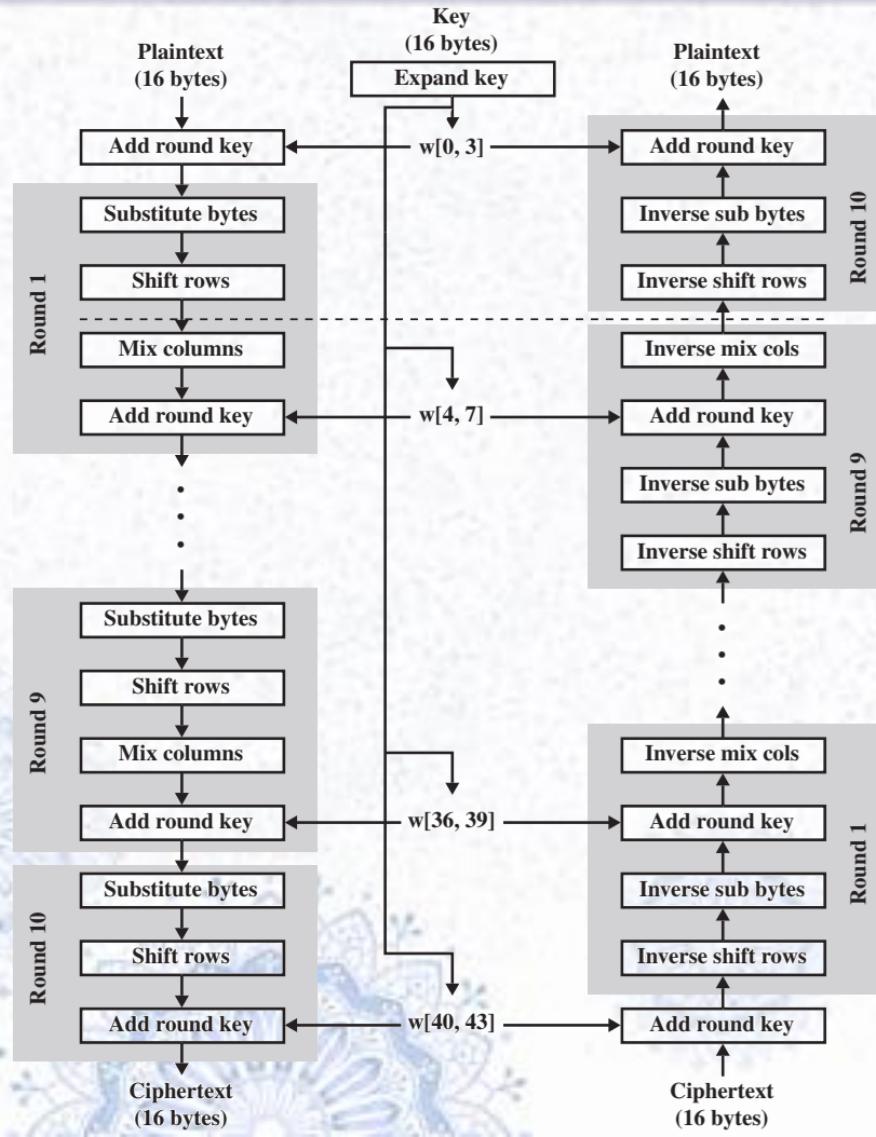
■ ترکیب خطی ستونهای ماتریس حالت با استفاده از ضرب ماتریسی  
■ اضافه نمودن کلید دور:

■ XOR ماتریس حالت با کلید دور

# رمزگذاری در AES-128



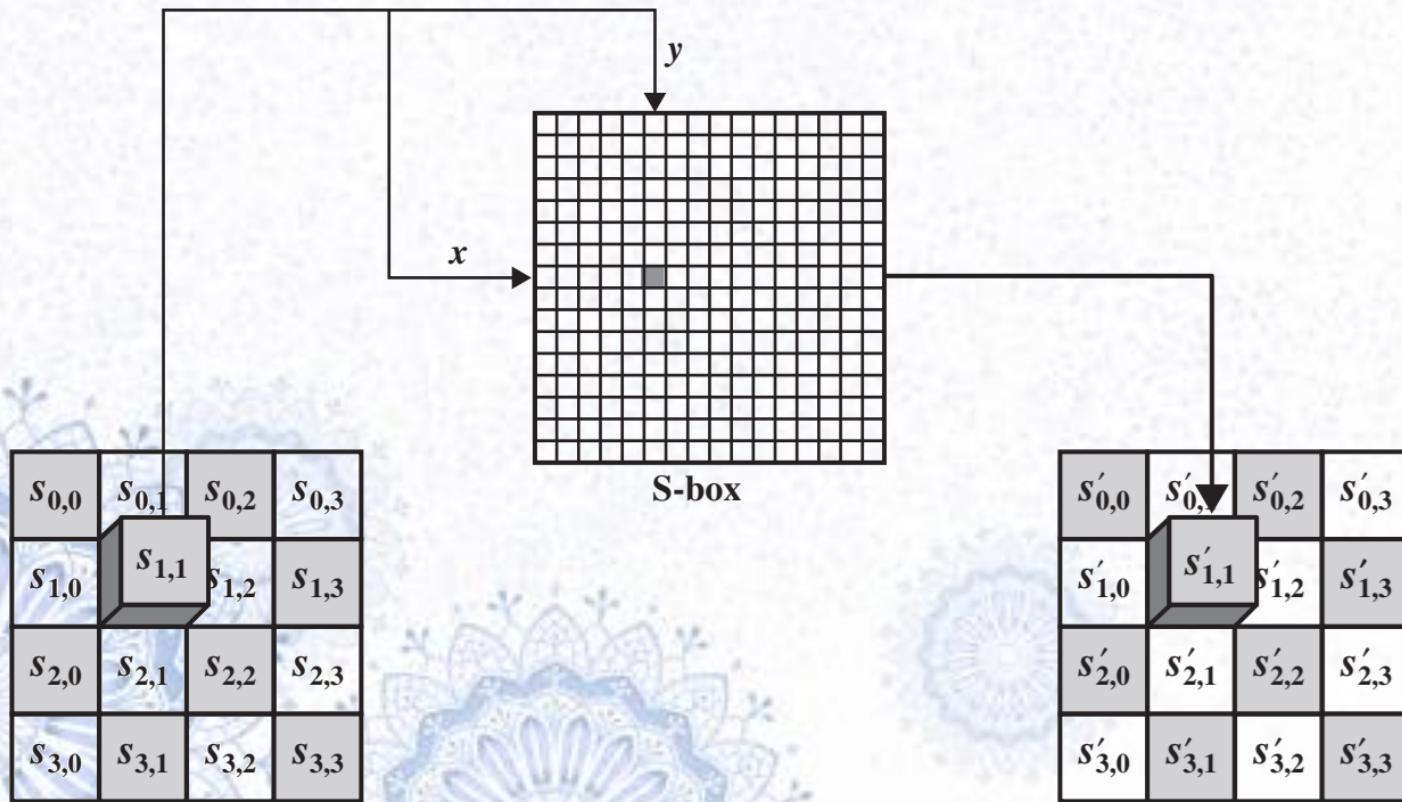
# رمزگذاری و رمزگشایی در AES-128



# جانشینی بایت‌ها (S-Box) در AES

نوعی تابع غیرخطی محاسبه می‌شود

توسط یک ماتریس  $4 \times 4$  بایت پیاده‌سازی می‌شود



# جانشینی بایتها (S-Box) در AES

- ورودی تابع سطر و ستون درایه جدول را معین می کند و  
▪ مقدار ذهنی شده در این درایه فروجی تابع است
- با داشتن یک عنصر از ماتریس حالت  
▪ سطر جدول = ۱۶ بیت سمت پایین عنصر  
▪ ستون جدول = ۱۶ بیت سمت راست عنصر
- برای رمزگشایی از جدول معکوس استفاده می شود

# جدول جانشینی در AES

	<i>y</i>																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
<i>x</i>	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

# جدول جانشینی معکوس در AES

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

# شیفت سطّری در AES

شیفت چرخشی به چپ که در آن:

سطر اول بدون تغییر

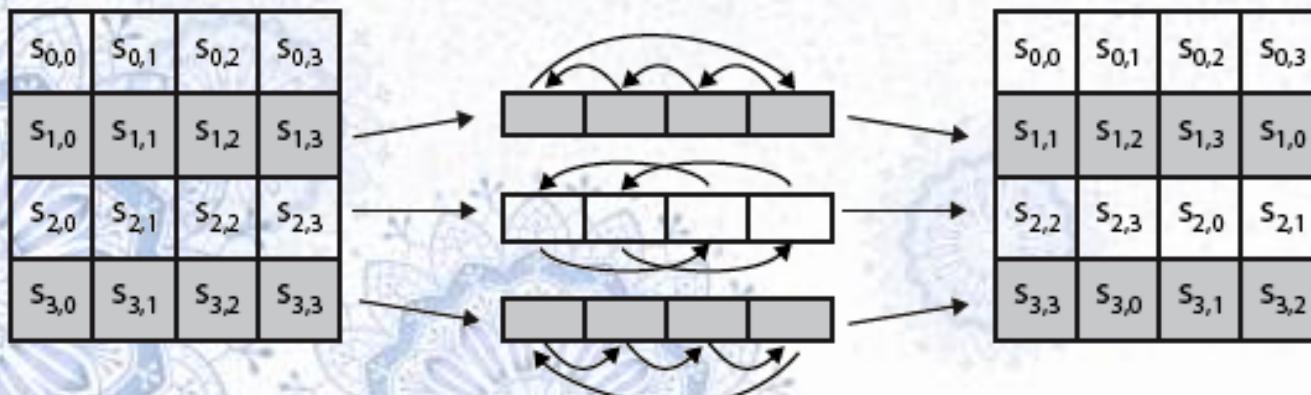
سطر دو<sup>م</sup> یک بایت شیفت چرخشی به چپ

سطر سوم<sup>م</sup> دو بایت شیفت چرخشی به چپ

سطر چهارم<sup>م</sup> سه بایت شیفت چرخشی به چپ

در رمزگشایی، شیفت چرخشی به راست انجام می‌شود

چون داده به صورت ستونی در ماتریس حالت ذخیره شده، این مرحله در واقع یک جایگشت است



# ترکیب ستونها در AES

☞ هر ستون جداگانه پردازش می شود

☞ هر بایت ستون با یک ترکیب فقط از بایتهای آن ستون جایگزین می شود

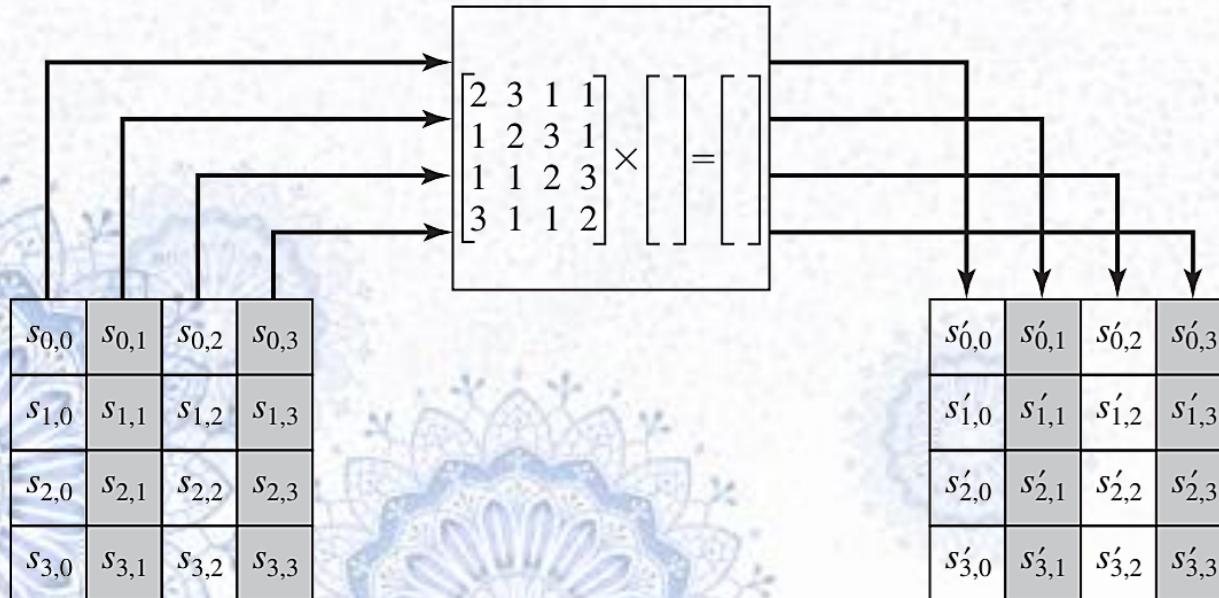
☞ با ضرب یک ماتریس در کل ستون

☞ ضرب و جمع، ضرب و جمع عادی نیستند.

☞ ضرب و جمع در میدان متناهی با اندازه  $2^8$ : مشابه ضرب و جمع چند جمله‌ای ها

☞ هدف: برگشت پذیر بودن

☞ برای رمزگشایی از ماتریس دیگری در ضرب استفاده می شود

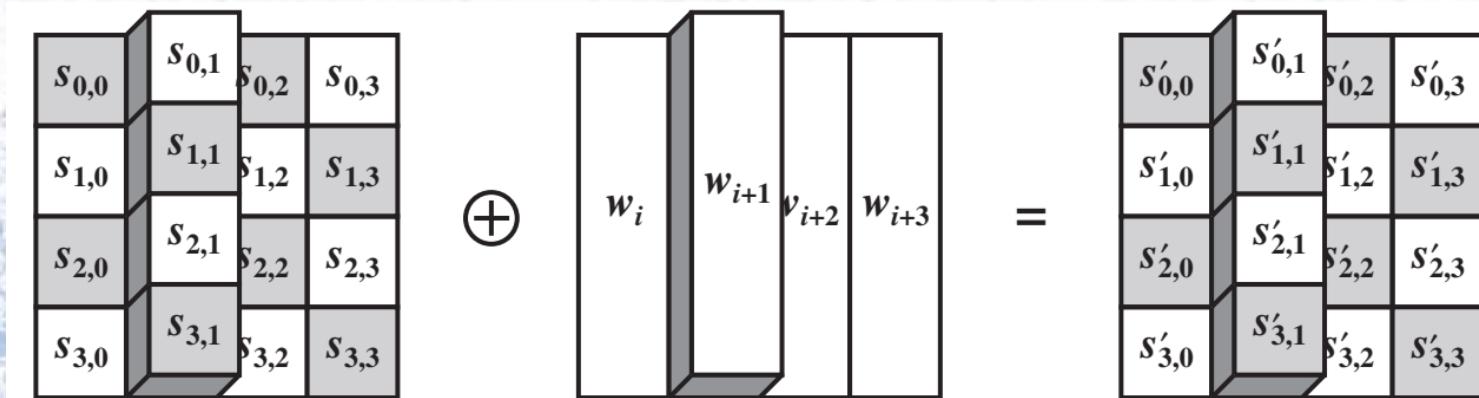


# افزودن کلید دور در AES

• ماتریس حالت با کلید دور XOR می‌شود

• به صورت ستونی انجام می‌شود

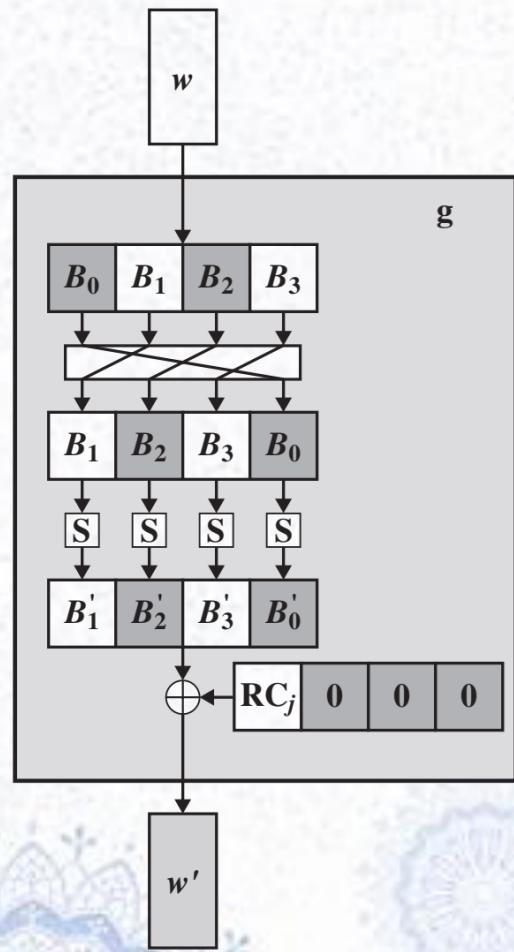
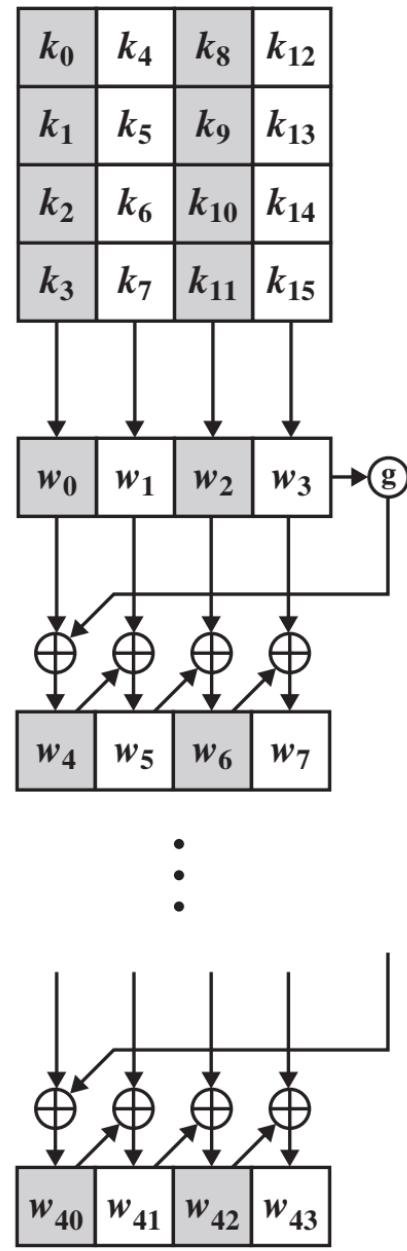
• برای رمزگشایی نیز همین عمل انجام می‌شود



# بسط کلید در AES-128

- یک کلید ۱۲۸ بیتی (۱۶ بایتی) دریافت می‌کند و آن را به یک آرایه ۴x۴ عنصره (از کلمات ۴x۴ بیتی) بسط می‌دهد
- شروع: کپی کلید در ۴ عنصر (کلمه) اول آرایه
- تکرار: تولید هر عنصر (کلمه  $w[i]$ ) بر اساس  $w[i-4]$  و  $w[i-1]$
- عناصر موجود در درایه‌های مضرب ۴ با تابع پیچیده  $g$  محاسبه می‌شوند

# بسط کلید در AES



# بسط کلید در AES

تابع پیچیده  $g$  شامل توابع فرعی یر است:

شیفت چرفشی به چپ به اندازه یک بایت (RotWord) 

جانشینی هر بایت بر اساس جدول S-Box مورد استفاده در (مزگذاری) SubWord 

ترکیب XOR مقدار حاصل از انجام اعمال ۱ و ۲ با مقدار ثابت Rcon[i/4] 

$$Rcon[i/4] = (RC[i/4], 0, 0, 0) \quad \text{$$

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

# امنیت AES

بهترین حمله به AES-128 پیچیدگی ۲<sup>۱۲۶.۱</sup> دارد  
عملأً معادل آزمون جامع

در مقایسه با DES:

- اگر بتوان کلید DES را از طریق آزمون جامع در یک ثانیه بازیابی کرد
- یعنی در هر ثانیه  $2^{56}$  کلید را امتحان کرد
- این ماشین کلید AES را در  $10^{14} \times 10^5$  سال بازیابی می‌نماید

# جنبه های پیاده سازی AES

- 锁 قابلیت پیاده سازی روی پردازنده های ۸ بیتی و ۳۲ بیتی
- 锁 قابلیت پیاده سازی کارا روی پردازنده های ۳۲ بیتی (علت اصلی انتخاب)
- 锁 همه اعمال با شیفت، XOR و استفاده از جداول look-up قابل انجام است
- 锁 اینتل در سال ۲۰۰۸ مجموعه دستورالعملهای AES را به CPU های خود افزود
- 锁 افزایش چشمگیر سرعت AES

دستورالعمل	توصیف
<b>AESENC</b>	اجرای یک دور عادی از رمزگذاری AES
<b>AESENCLAST</b>	اجرای دور آخر از رمزگذاری AES
<b>AESDEC</b>	اجرای یک دور عادی از رمزگشایی AES
<b>AESDECLAST</b>	اجرای دور آخر از رمزگشایی AES
<b>AESKEYGENASSIST</b>	کمک در تولید کلید دور AES
<b>AESIMC</b>	کمک در عملیات Inverse Mix Columns
<b>PCLMULQDQ</b>	ضرب بدون رقم نقلی (عملیات در میدانهای متناهی)

lock وجود مجموعه دستورالعمل مشابه برای سایر معماری های CPU نظیر ARM

# فهرست مطالب

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- **رمزهای متقارن معروف**
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# سایر رمزهای متقاضی معروف...

GOST

FEAL

IDEA

Blowfish

Twofish

Threefish

RC2

RC5

RC6

MARS

Serpent

Camellia

CAST-128

CAST-256

# مقایسه سرعت الگوریتم‌ها (CBC—سبک BestCrypt)

Algorithm	Encrypt (MB/s)	Decrypt (MB/s)
AES (Rijndael) Hardware	1638.4	1638.4
RC6	806.5	1024
IDEA	806.5	806.5
BLOWFISH-448	641	641
TWOFISH	641	641
BLOWFISH-128	531.9	531.9
AES (Rijndael)	531.9	485.4
CAST	458.7	458.7
SERPENT	290.7	295.9
DES	267.4	266
GOST	245.1	246.3
3DES	96.9	94

# فهرست مطالب

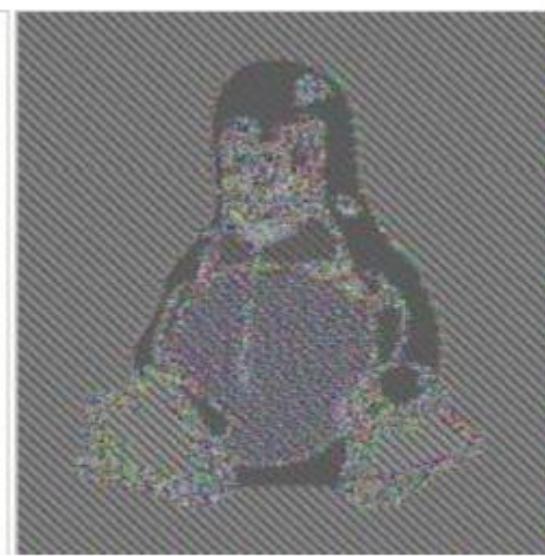
- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- رمزهای جریانی

# انگیزه

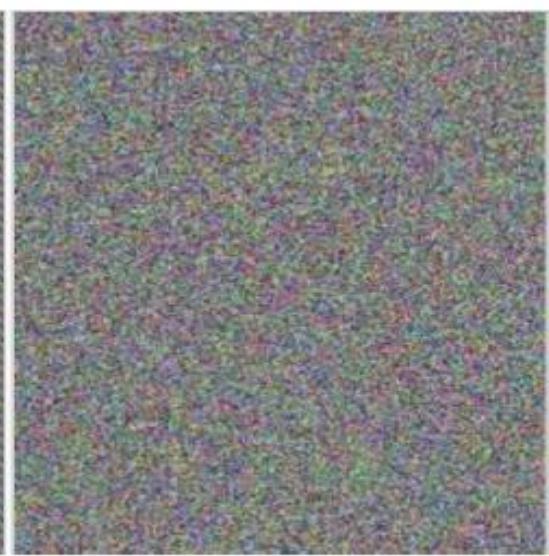
- ❖ رمزنگاری یک پیام طولانی چگونه باشد؟
- ❖ طول متن آشکار بسیار بزرگتر از اندازه قالب
- ❖ قطعه بندی پیام، و رمزنگاری هر قالب به طور مستقل.
- ❖ نتیجه:



پیام



رمز بد!



رمز خوب

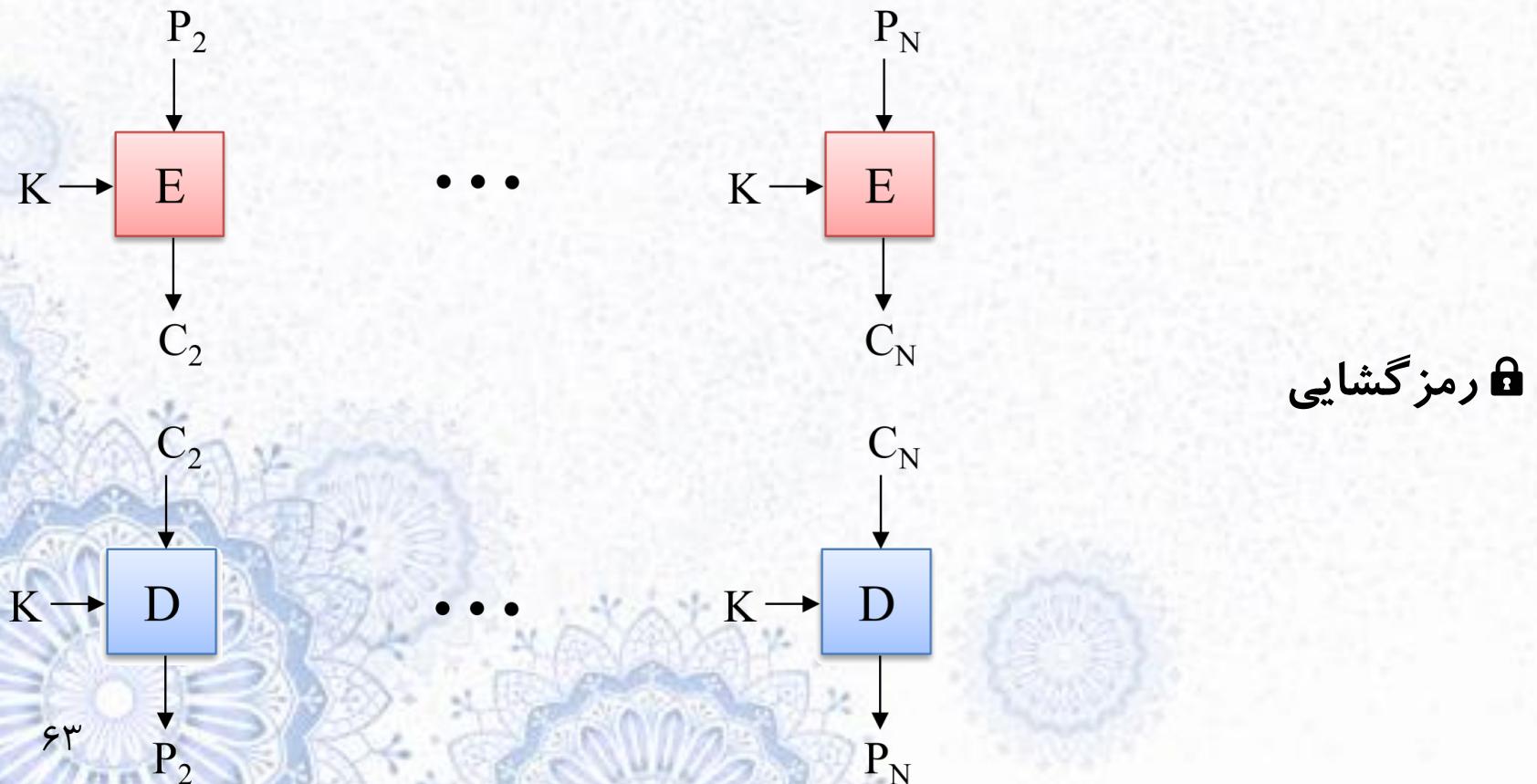
# سبک‌های کاری (Modes of Operation)

- امروزه سبک‌های کاری با توجه به امنیت قابل اثبات طراحی می‌شوند
- سبک‌های کاری می‌توانند از رمزهای قالبی دلخواه استفاده کنند
- برخی سبک‌های کاری پر اهمیت عبارتند از:

- ECB: Electronic Codebook**
- CBC: Cipher Block Chaining**
- CTR: Counter Mode**
- CFB: Cipher Feed Back**
- OFB: Output Feed Back**

# سُبک کاری ECB

• رمزگذاری و رمزگشایی مستقل هر قطعه با کلید  
• رمزگذاری



# بررسی سبک کاری ECB

اشکال اساسی :

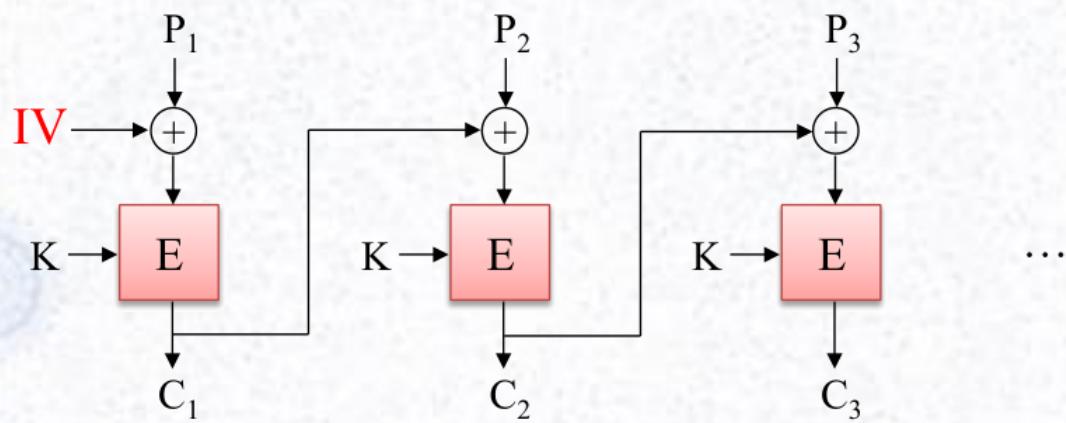
هر متن آشکار به ازاء کلید ثابت همیشه به یک متن (مز شده نگاشته) می شود

حمله کننده می تواند دریابد که پیام های یکسان ارسال شده اند

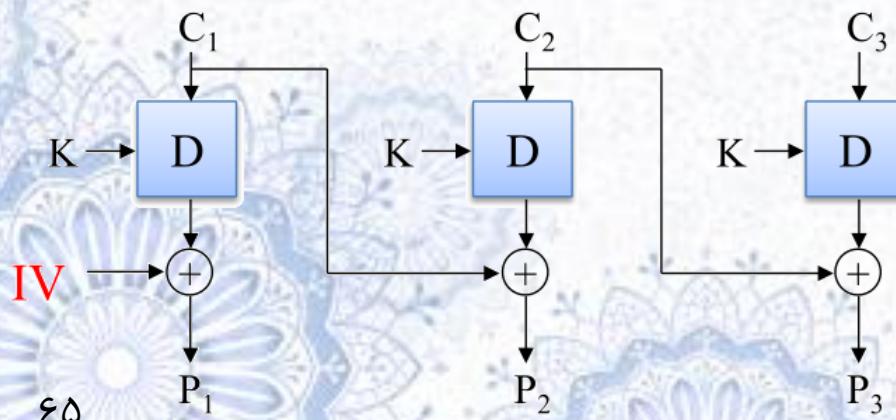
این سبک امن محسوب نمی شود حتی اگر از یک رمز قالبی قوی استفاده کنیم

ECB مثالی از مواردی است که علی رغم بهره برداری از عناصر مرغوب، کیفیت نهايی دلخواه نیست

# سېك کاري CBC



رمزگذاري



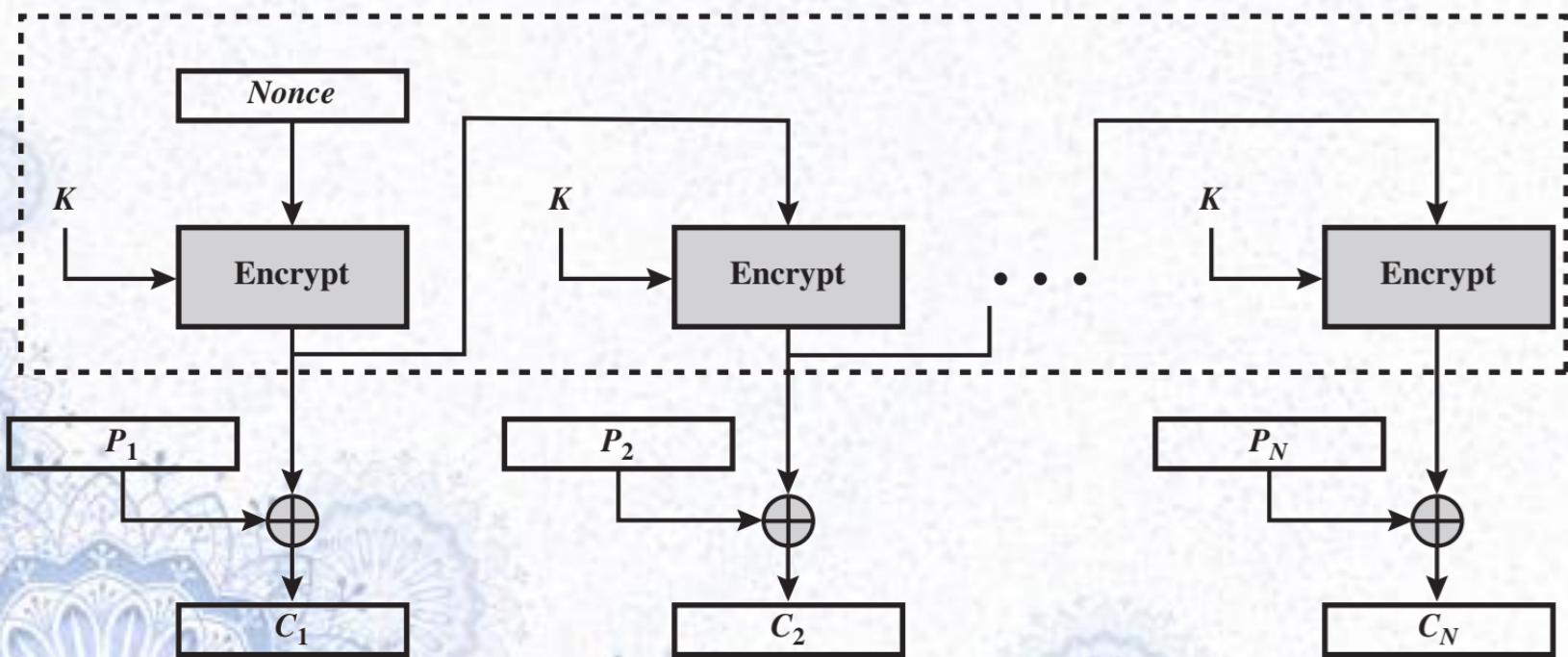
رمزگشايي

# سبک کاری CBC

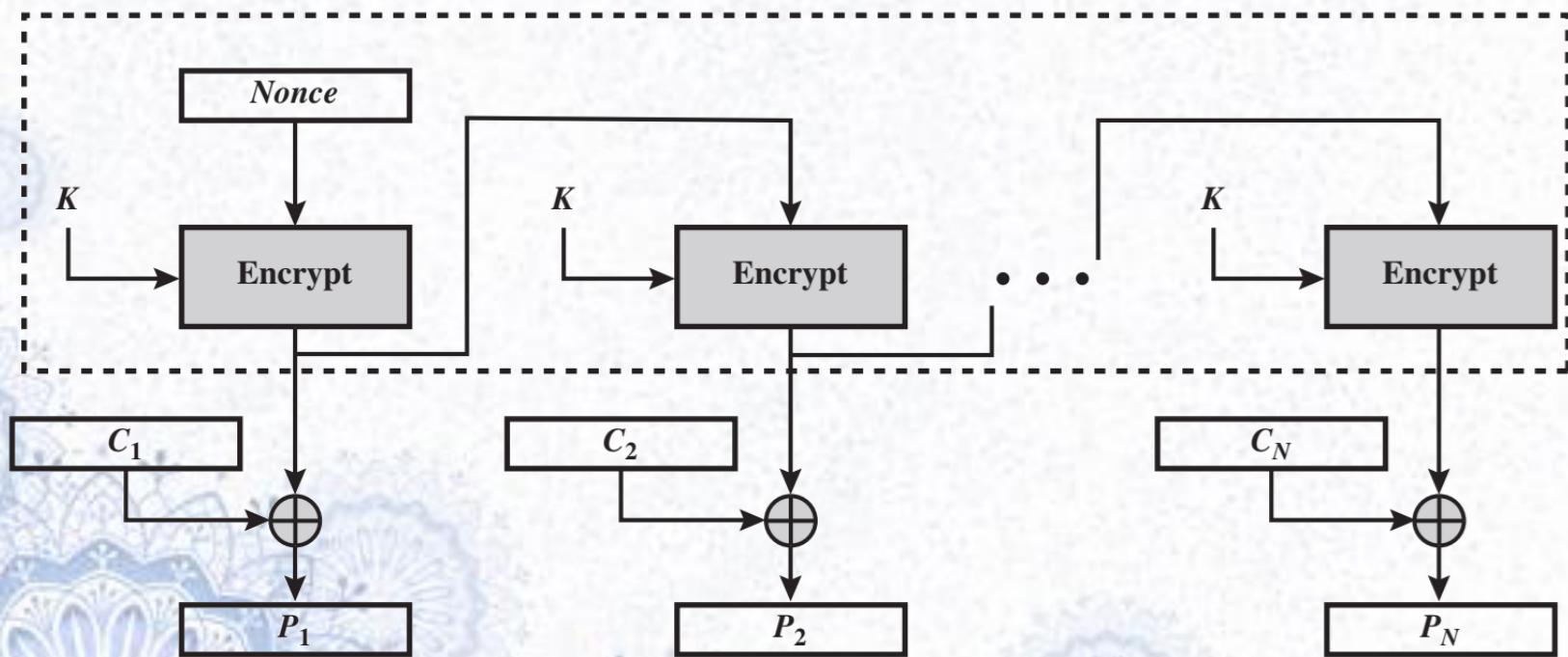
- این سبک (و برحی سبک‌های دیگر) از یک بهره می‌گیرد
- مقدار IV در هر بار رمزگذاری به صورت تصادفی تغییر می‌کند
- همراه با متن رمز شده ارسال می‌شود
- ارسال IV به صورت آشکار تاثیری در امنیت رمز ندارد
- هر متن آشکار به ازای کلید ثابت هر بار به یک متن رمز شده متفاوت تبدیل می‌شود
- زیرا مقدار IV تغییر می‌نماید

- ویژگی‌های این سبک
- IV باید کاملاً غیر قابل پیش‌بینی باشد
- عملیات رمزگذاری قابل موازی سازی نیست
- عملیات رمزگشایی قابل موازی سازی است

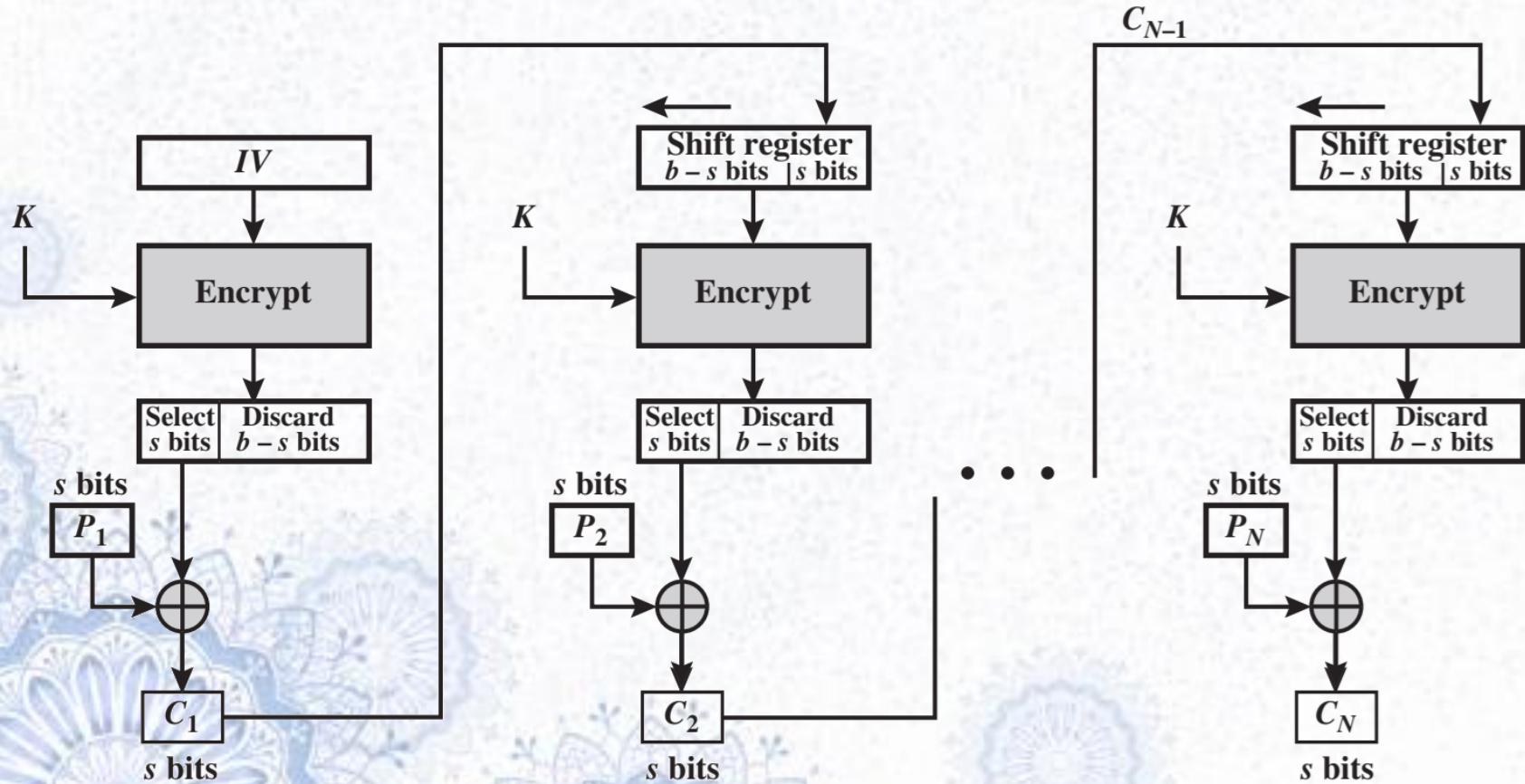
# سُبک کاری OFB – رمزگذاری



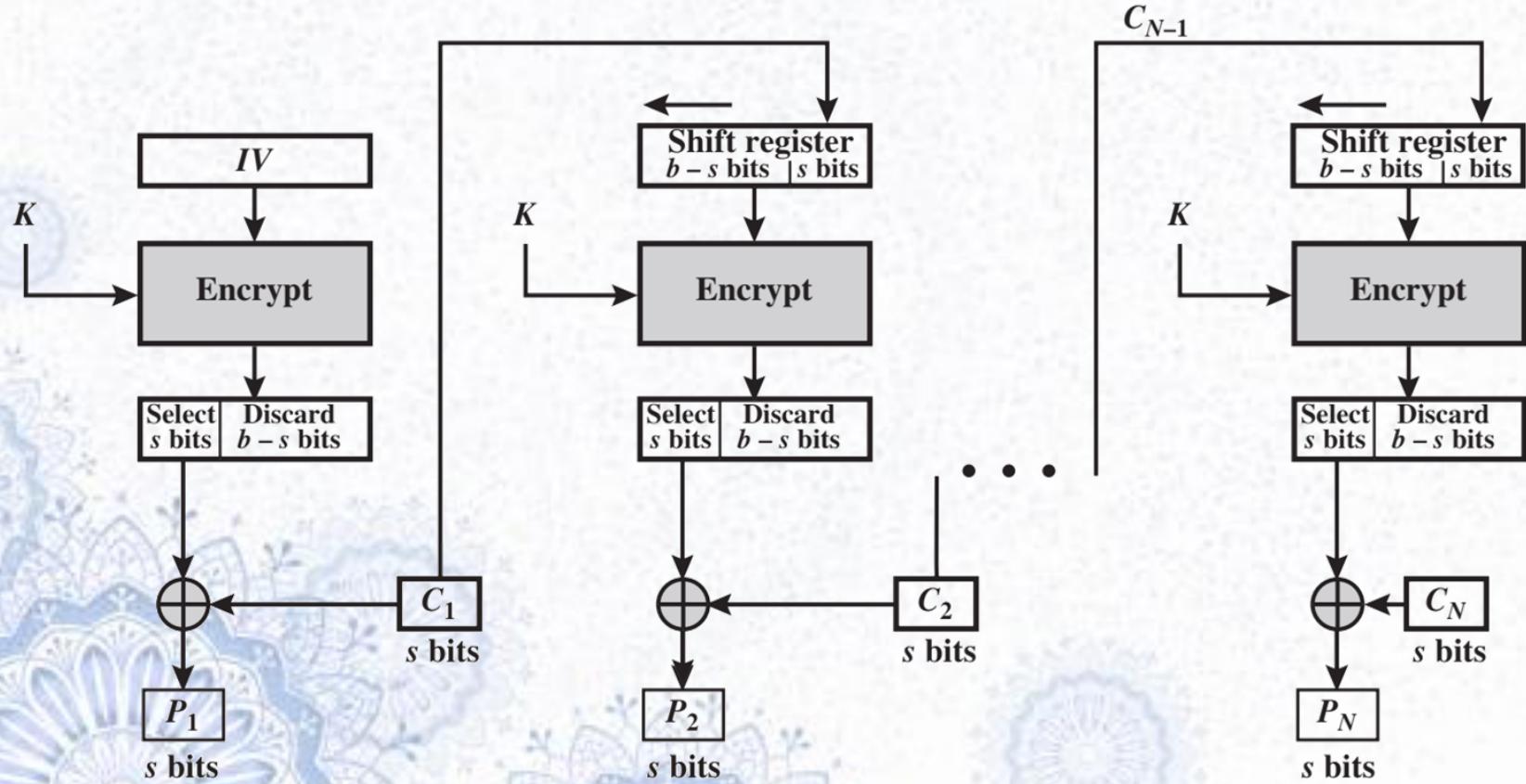
# سُبک کاری OFB – رمزگشایی



# سُبک کاری CFB – رمزگذاری



# سُبک کاری CFB – رمزگشایی



# مقایسه OFB و CFB

• موارد استفاده OFB و CFB:

• به عنوان رمزگشایی

• در کاربردهای بی‌درنگ (مثل SSH)

• عیب CFB: انتشار خطای انتقال

• این عیب OFB ندارد

• این عیب در کاربردهای تصدیق هویت، حسن محسوب می‌شود!

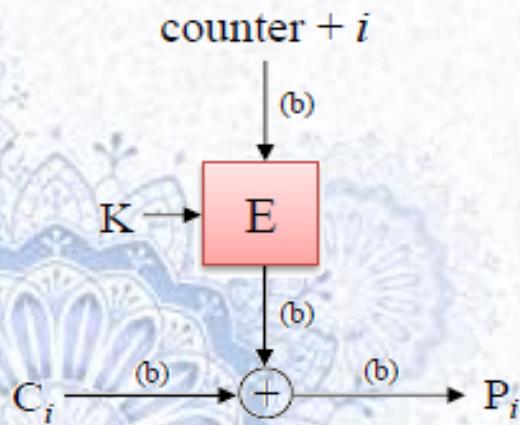
• عیب OFB: رمزگشایی قابل موازی سازی نیست

• این مشکل CFB ندارد

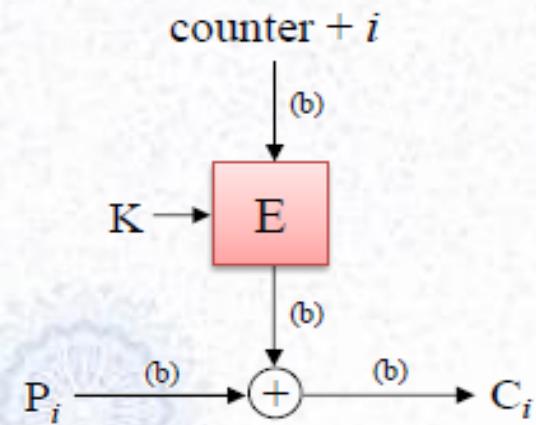
# سُبک CTR کاری

- شمارنده ای با طول قطعه(bیت) به تصادف انتخاب می شود
- برای هر قطعه به شمارنده یک واحد اضافه می شود (در پیمانه<sup>b</sup>2)

رمزگشایی



رمزگذاری



# بررسی سُک کاری CTR

 ملزومات امنیتی:

 مقادیر شما (نده، در بازه طول عمر کلید، باید مجزا باشند

 رمزگذاری و رمزگشایی:

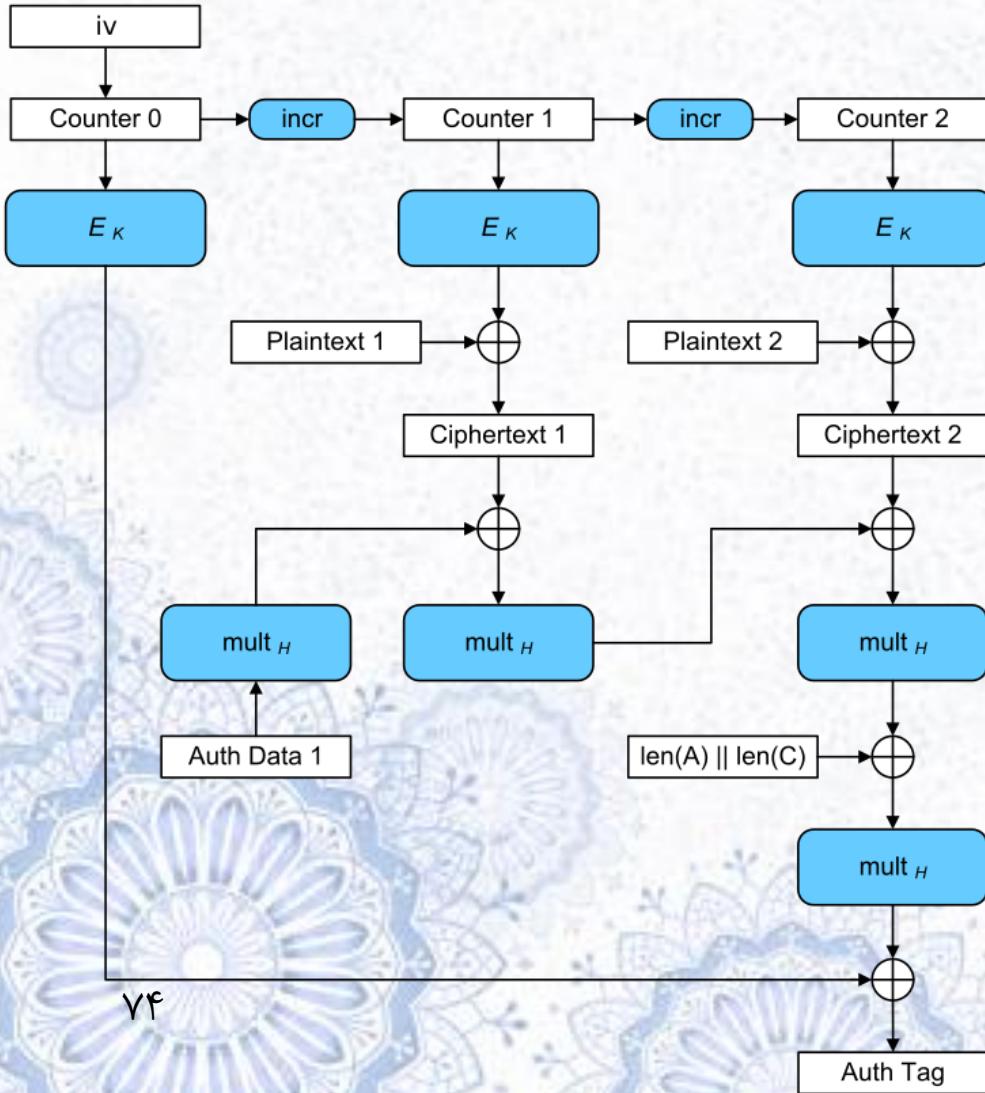
 عملیات (رمزگذاری و رمزگشایی قابل موازی‌سازی است

 برای شروع عملیات (رمزگذاری نیازی به متن آشکار نیست

 برای شروع عملیات (رمزگشایی نیازی به متن رمز نیست

 امکان پیاده‌سازی کارا به صورت سفت افزایی و نرم افزایی

# سُبک کاری GCM



CTR شبیه به سُبک  
امکان رمزگاری و احراز هویت  
همزمان  
علاوه بر متن رمز یک برجسب  
برای احراز صحت پیام تولید  
می‌شود

# مقایسه کاربرد انواع سبک های کاری

سبک کاری	کاربرد
ECB (Electronic Code Book)	کلاً امن نیست و نباید استفاده شود!
CBC (Cipher Block Chaining)	ارسال قطعه-گرای هر گونه داده تصدیق صحت
OFB (Output Feed Back)	ارسال جریانی بر روی کانال نویزی (مانند ارتباطات ماهواره‌ای)
CFB (Cipher Feed Back)	ارسال جریانی هر گونه داده تصدیق صحت
CTR (Counter)	ارسال قطعه-گرای هر گونه داده مناسب برای ارسال با سرعت بالا

# سایر سبک های کاری

امروزه تعداد زیادی سبک کاری برای اهداف مختلف ابداع شده اند

(مزنگاری دیسک سفت)

(مزنگاری تصدیق صحت شده )

...

برخی سبک های رمزنگاری معروف دیگر :

CCM, CMC, CWC, EAX, EME, GCM, IACBC, IAPM, LRW, OCB, XCBC, XEX, XTS

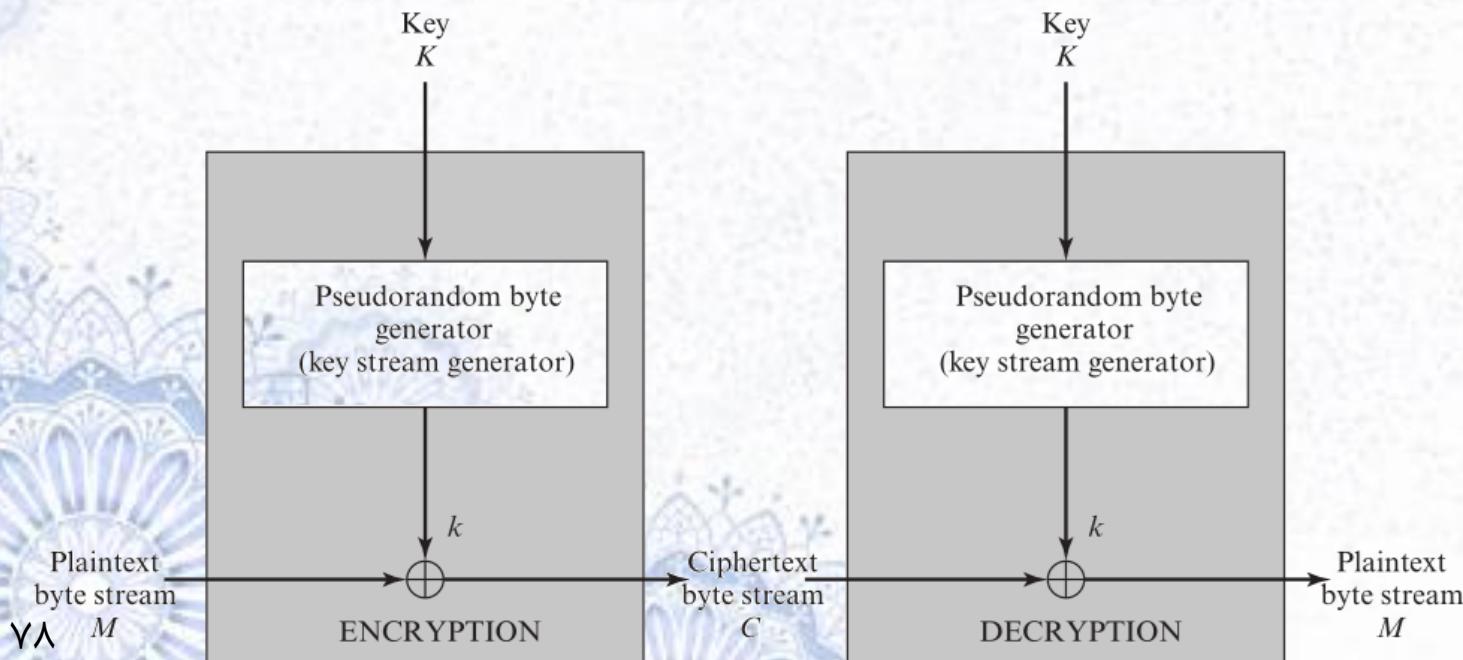
# فهرست مطالب

- رمزهای قالبی و جریانی
- ساختارهای SPN و فایستل و ویژگیهای آشفتگی و پخش
- استاندارد رمزگذاری داده (DES)
- الگوریتمهای رمز 2DES و 3DES
- استاندارد رمزگذاری پیشرفته (AES)
- رمزهای متقارن معروف
- سبک های کاری رمزهای متقارن
- **رمزهای جریانی**

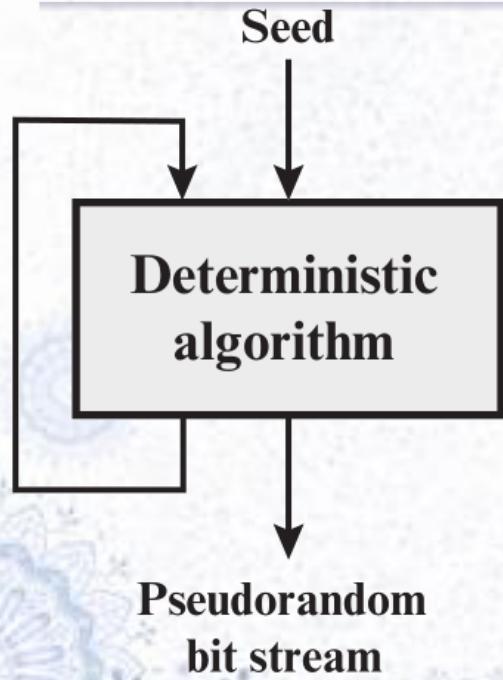
# رمزهای جریانی

## ایده اصلی

- OTP از دنباله تصادفی واقعی (TRN) استفاده می کند
- تولید و به اشتراک گذاری دنباله تصادفی واقعی سفت است
- از دنباله شبیه تصادفی (PRN) برای رمزنگاری استفاده کنیم
- دو طرف بتوانند از یک کیلد مشترک کوتاه یک دنباله تصادفی طولانی درست کنند



# دنباله‌های شبیه تصادفی



تولید دنباله ای از اعداد که حدالامکان خواص آماری تصادفی داشته باشد

با یک الگوریتم قطعی (Deterministic) و وجودی الگوریتم یک مقدار اولیه به نام seed ( تنها منبع تصادفی )

دنباله شبیه تصادفی فقط به seed بستگی دارد  
با دانستن تنها seed می توان دنباله را مجددا تولید کرد

خواص آماری مطلوب یک دنباله تصادفی :

یکنواختی: فراوانی بیت‌های یک و صفر یکسان باشد  
(uniformity)

استقلال: هیچ زیردنباله از دوی زیردنباله های دیگر قابل استنتاج نباشد (independence)

▪ تخمین غیر قابل پیش بینی بودن (Unpredictability)

▪ بررسی آن سفت است

# رمز RC4

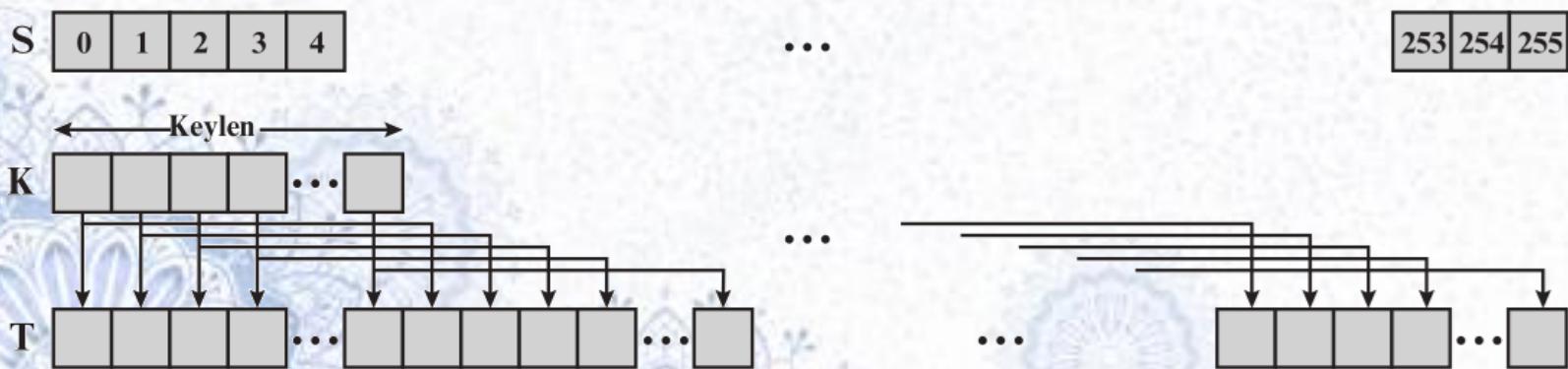
یک رمز بسیار ساده توسط Ron Rivest  
بسیار کارا

برای هر بایت ۸ تا ۱۶ عمل ماشین نیاز است  
برپایه دنباله شبه تصادفی تولید شده با جایگشت تصادفی  
با طول دوره بیش از  $10^{100}$   
طول کلید متغیر بین ۱ تا ۲۵۶ بایت  
اخیراً آسیب‌پذیری برای این روش کشف شده که امکان کشف متون آشکاری که  
مکرراً با این روش رمز شده اند را ممکن می‌کند

# مقداردهی های اولیه در RC4

```
/* Initialization */  
for i = 0 to 255 do  
    S[i] = i;  
    T[i] = K[i mod keylen];
```

```
/* Initial Permutation of S */  
j = 0;  
for i = 0 to 255 do  
    j = (j + S[i] + T[i]) mod 256;  
    Swap (S[i], S[j]);
```



# رمز جریانی RC4

```
/* Stream Generation */  
i, j = 0;  
while (true)  
    i = (i + 1) mod 256;  
    j = (j + S[i]) mod 256;  
    Swap (S[i], S[j]);  
    t = (S[i] + S[j]) mod 256;  
    k = S[t];
```

