



Shahid Beheshti  
University

# رمزنگاری

هادی سلیمانی

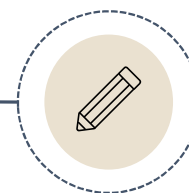
پژوهشکده فضای مجازی دانشگاه شهید بهشتی

- اجازه‌ی ایجاد نسخه‌های دیجیتالی جدید براساس بخشی یا تمام مطالب این اسلاید بدون پرداخت هزینه اعطا می‌شود، مشروط بر این‌که:
- فقط به‌منظور و در راستای استفاده‌ی آموزشی (شخصی و یا کلاسی) ساخته شده باشند و برای کسب هرگونه سود و یا مزیت تجاری استفاده نشوند.
- نسخه‌های جدید حاوی ارجاع مستقیم به نام تهیه‌کننده اسلاید (هادی سلیمانی) و محل کار وی (پژوهشکده فضای مجازی دانشگاه شهید بهشتی) باشند.
- مجموعه‌ی حاضر براساس نظرات ارزشمند دانشجویان (سابق) دانشگاه شهید بهشتی و همکاران محترم تهیه شده است که از تمام آن‌ها قدردانی می‌شود؛
- (به‌خصوص خانم‌ها **سارا زارعی و فاطمه عزیزی** نقش مهمی را در تهیه نسخه‌ی نهایی بر عهده داشته‌اند. خانم مهندس زارعی علاوه بر کمک در آماده‌سازی نسخه‌ی فعلی اسلایدها، در تصحیح اشتباهات نسخه‌ی قبلی و همچنین تکمیل و بازتعریف محتوای درس‌ها بسیار تاثیرگذار بوده‌اند).
- برای مشاهده‌ی اسلایدها و ویدئوهای تدریس این درس به آدرس زیر مراجعه فرمایید:

[http://facultymembers.sbu.ac.ir/h\\_soleimany/cryptography-course/](http://facultymembers.sbu.ac.ir/h_soleimany/cryptography-course/)

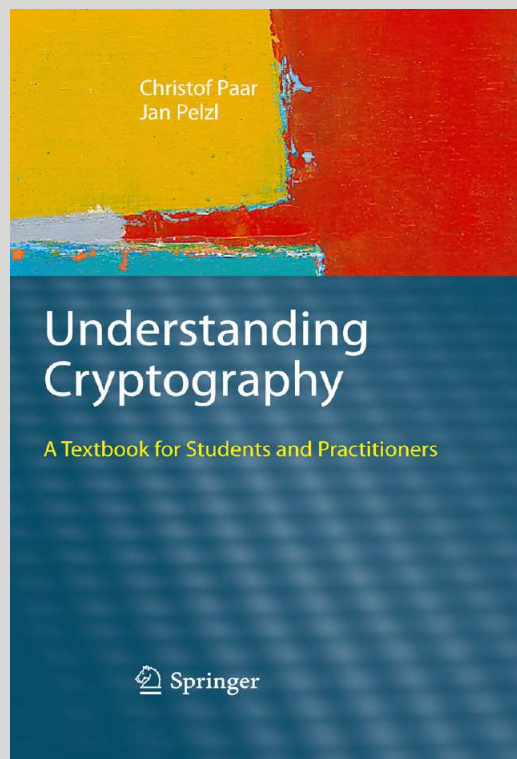
## درس پنجم

### رمزهای قالبی




## ■ معرفی مرجع

### رمزهای قالبی



Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.

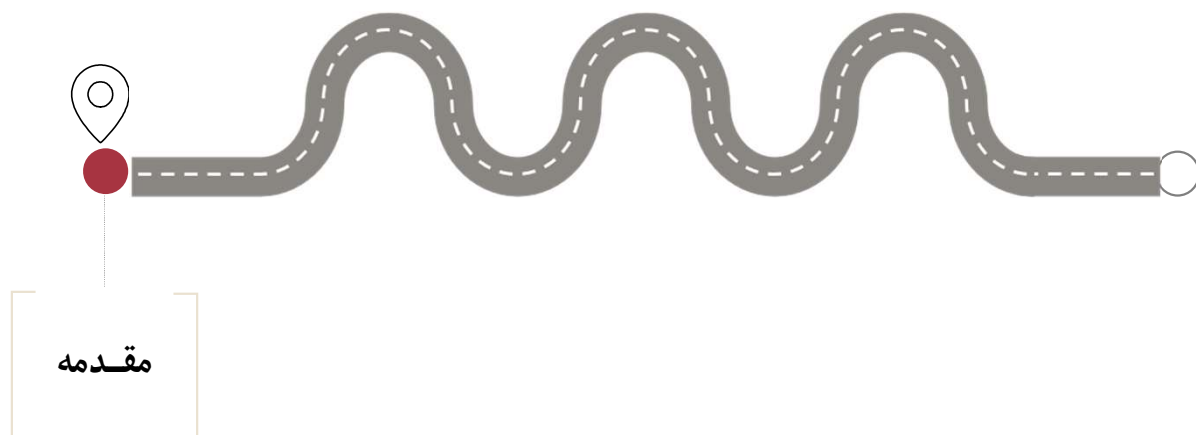
مثال‌ها و تصاویر این بخش از درس از کتاب مرجع است (در غیر این صورت مرجع شکل ذکر شده است). 

## ■ فهرست عناوین درس

### رمزهای قالبی

- مقدمه
- آشفته‌سازی و پراکنش
- رمزهای قالبی تکرارپذیر
- رمزهای قالبی فیستلی
- رمزهای قالبی SPN
- مقایسه‌ی رمزهای جریانی و رمزهای قالبی
- جمع‌بندی مطالب





## ■ رمز قالبی

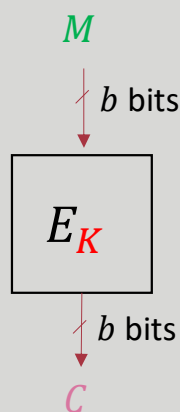
### (Block Cipher)

- الگوریتم رمزگذاری  $E_K$ ، یک متن اصلی  $b$  بیتی را براساس کلید الگوریتم  $(K)$  به یک متن رمزشده  $b$  بیتی تبدیل می‌کند.

- الگوریتم رمزگشایی  $D_K$ ، معکوس الگوریتم رمزگذاری  $E_K$  است، یعنی:

$$D_K(E_K(M)) = M$$

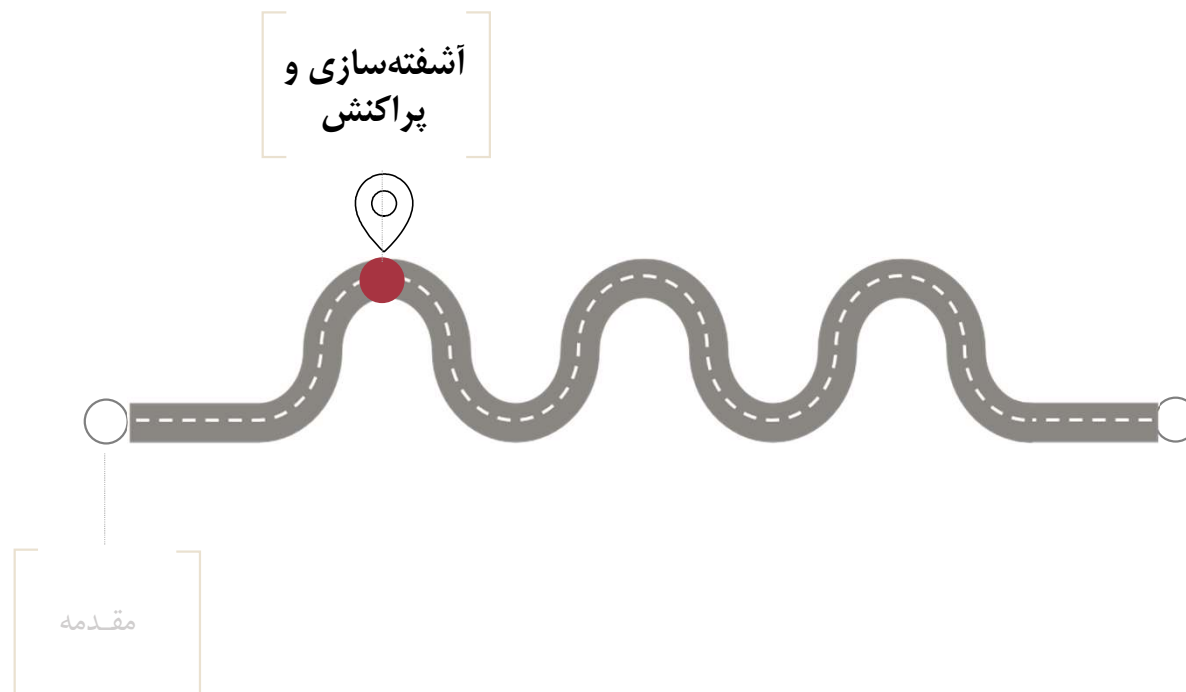
- بنابراین الگوریتم رمزگذاری  $E_K$  به ازای هر کلید، باید یک جایگشت  $b$  بیتی معکوس‌پذیر باشد.
- از آنجایی که ورودی به صورت قالب‌های  $b$  بیتی است، این دسته از الگوریتم‌های رمزنگاری را رمز قالبی نامیده‌اند.



$$E: \{0,1\}^b \times \{0,1\}^k \rightarrow \{0,1\}^b$$

- تعداد جایگشت‌های  $b$  بیتی ممکن  $2^{b(b-1)/2} \approx 2^{b!}$  است که به مراتب از تعداد جایگشت‌های یک رمز قالبی  $2^k$  بیشتر است (مطلقاً قابل قیاس نیست!).
- از منظر امنیتی، هدف کلی از طراحی یک رمز قالبی انتخاب کاملاً تصادفی  $2^k$  جایگشت از  $2^{b!}$  حالت ممکن است.
- به عبارت دیگر، ویژگی‌های رمز قالبی باید به ازای هر **کلید** دلخواه کاملاً تصادفی به نظر برسد.
- ساخت الگوریتم‌های رمزنگاری قالبی متأثر از دو مفهوم کلیدی است که توسط شانون ارائه شدند: آشفته‌سازی و پراکنش.





## ■ دو مفهوم کلیدی در مقاله شانون



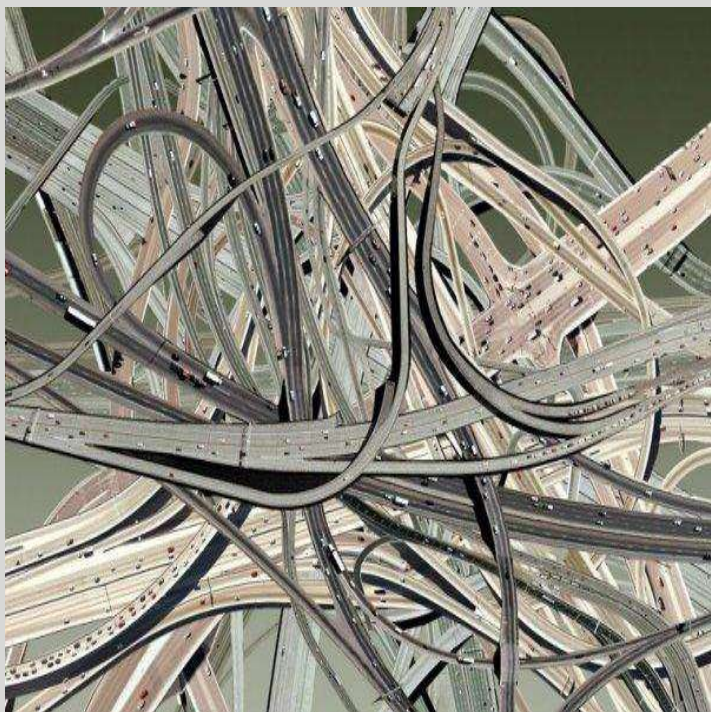
❖ Claude E. Shannon:

Two methods suggest themselves for frustrating a statistical analysis. These we may call the methods of **diffusion** and **confusion**.

## ■ آشفته‌سازی

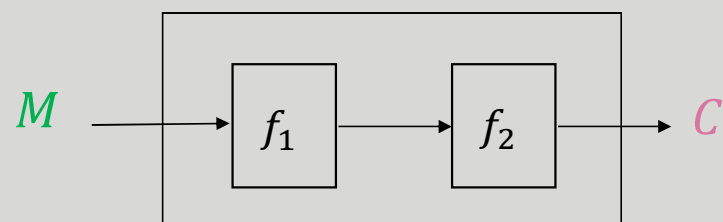
(Confusion)

- تعریف اولیه‌ی شانون: رابطه‌ی بین **متن رمز شده** و **کلید** حتی‌المقدور پیچیده باشد.
- مفهوم: با داشتن تعداد زیادی **متن رمز شده**، نتوان **کلید** را کشف کرد.
- در برخی از متون علمی از آشفته‌سازی به عنوان «پیچیدگی رابطه‌ی **متن اصلی** و **متن رمز شده**» نیز یاد می‌شود.



## ■ آشفته‌سازی

### مثال (یک رمز آفینی ترکیبی)



$$\begin{aligned} C &= f_2 f_1(M) = A_2(A_1 M + B_1) + B_2 \\ &= A_2 A_1 M + (A_2 B_1 + B_2) \\ &= A_3 M + B_3 \end{aligned}$$

- فرض کنید که یک الگوریتم رمزنگاری از ترکیب دو تبدیل آفینی تشکیل شده باشد:

$$f_1(x) = A_1 x + B_1 \text{ و } f_2(x) = A_2 x + B_2$$

که مقادیر  $(A_1, B_1)$  و  $(A_2, B_2)$  کلید مخفی هستند.

- نتیجه: دو تبدیل آفینی معادل یک تبدیل آفینی هستند.
- بنابراین برای شکست سیستم می‌توان به جای پیدا کردن چهار مقدار  $(A_1, B_1)$  و  $(A_2, B_2)$ ، دو مقدار  $(A_3, B_3)$  را حدس زد و پیدا کرد (نقطه‌ی ضعف این الگوریتم).
- دلیل: وجود یک رابطه‌ی ساده بین ورودی و خروجی تابع!

- استفاده از توابع غیرخطی برای آشفته‌سازی ضروری است.
- راه‌کار معمول استفاده از توابع غیرخطی کوچک است که تحت عنوان جعبه‌های جانشانی (Substitution box) شناخته می‌شوند.
- تاکنون در بسیاری از رمزهای قالبی به کار رفته‌اند (نظیر رمزهای استاندارد AES و DES).
- به خاطر وجود کارهای نظری فراوان در این حوزه، امنیت آن‌ها به خوبی فهمیده می‌شود.
- راه‌کار دیگری که در برخی از رمزهای قالبی به کار رفته نیز استفاده از جمع پیمانه‌ای است.
- در پیاده‌سازی‌های نرم‌افزاری کارایی بهتری دارد، اما امنیت آن برای جامعه‌ی رمزنگاری به روشنی جعبه‌های جانشانی نیست.

## ■ پراکنش

(Diffusion)

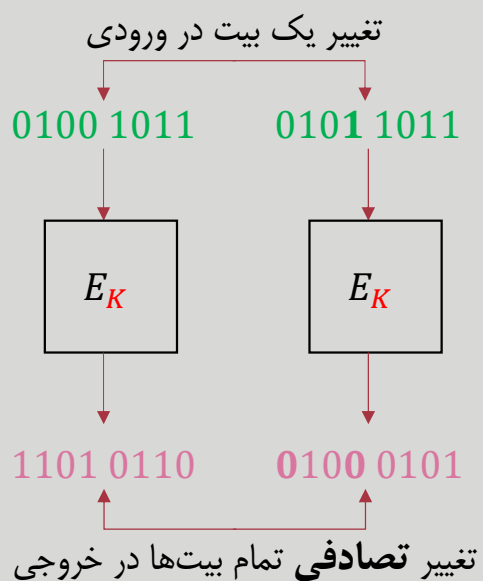
- تعریف شانون: از بین بردن هرگونه رابطه‌ی آماری بین دسته‌ای از **متون اصلی** و **متون رمز شده**.
- یعنی اگر **متون اصلی** با یکدیگر رابطه داشته باشند، **متون رمز شده**ی معادل آنها با یکدیگر رابطه‌ی آماری خاصی نداشته باشند.
- بنابراین، برای برآورده کردن ویژگی پراکنش، هر دسته‌ی خاص از **متون اصلی** باید بتوانند به تمامی  $2^b$  مقدار ممکن منتقل شوند.
- استفاده از توابع خطی نقش مهمی در ایجاد پراکنش دارد.

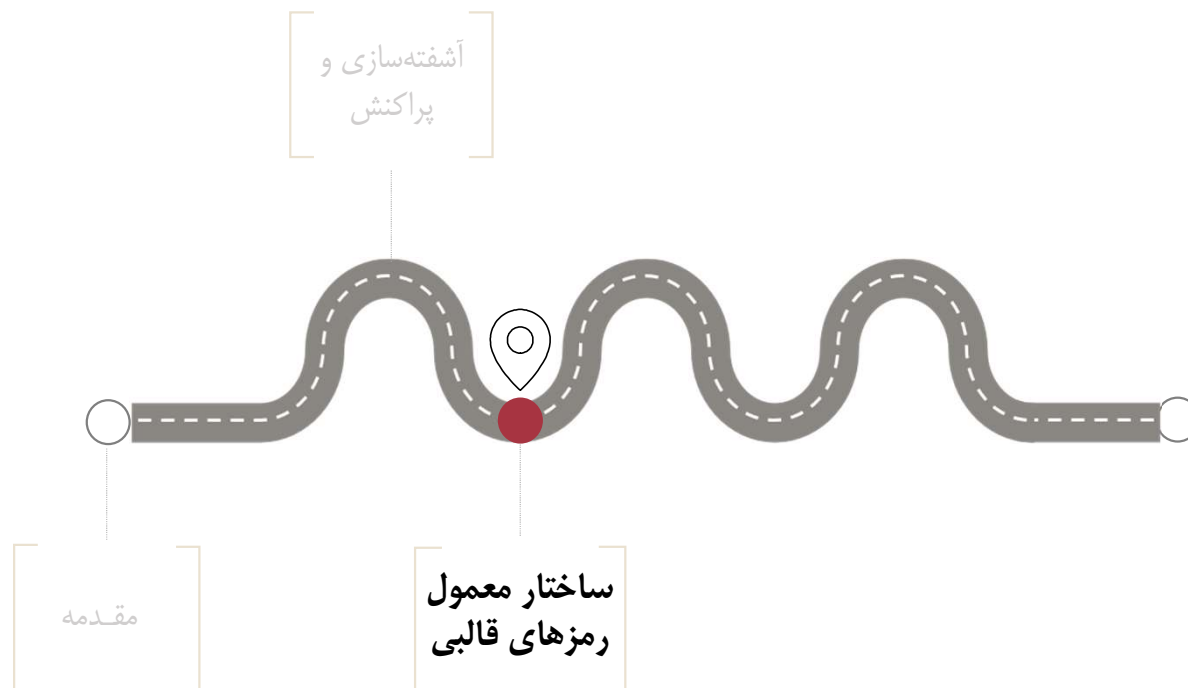


## ■ اثر بهمنی

### (Avalanche Effect)

- اثر بهمنی: بر اثر تغییر یک بیت از ورودی، هر بیت از خروجی با احتمال تقریباً ۵۰ درصد تغییر کند.
- **مفهوم:** با تغییر کم **متن اصلی**، **متن رمز شده** به صورت کاملاً تصادفی تغییر کند.
- یا به عبارت دیگر، یک دسته‌ی خاص از **متون اصلی** منجر به یک دسته‌ی خاص از **متون رمز شده** نشوند.
- معیار بهمنی اکید (Strict Avalanche Criterion): با تغییر یک بیت از ورودی، هر بیت از خروجی با احتمال دقیقاً ۵۰ درصد تغییر کند.





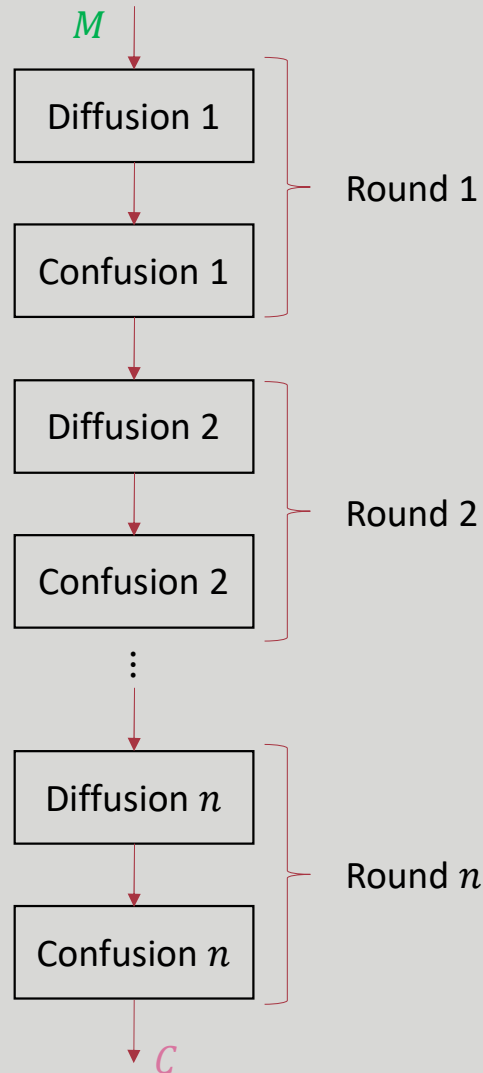


## ■ استفاده‌ی هم‌زمان از جانشانی و جابه‌جایی

- شانون: روش‌های جانشانی و جابه‌جایی باید به صورت هم‌زمان و متناوب استفاده شوند.
- به همین علت، رمزهای قالبی عموماً با استفاده از تکرار یک عملیات به نام دور (Round) ساخته می‌شوند.
- هر دور شامل تعدادی تابع (ساده) خطی و غیرخطی است.
- عملیات دور باید یک‌به‌یک باشد تا رمزگشایی نیز ممکن باشد.
- معمولاً طراحان ساختار دورها را یکسان (یا بسیار شبیه به هم) در نظر می‌گیرند تا امکان پیاده‌سازی الگوریتم‌ها با کارایی بالاتری وجود داشته باشد.
- به این دسته از رمزها، رمزهای قالبی تکرارپذیر (Iterated) گفته می‌شود.

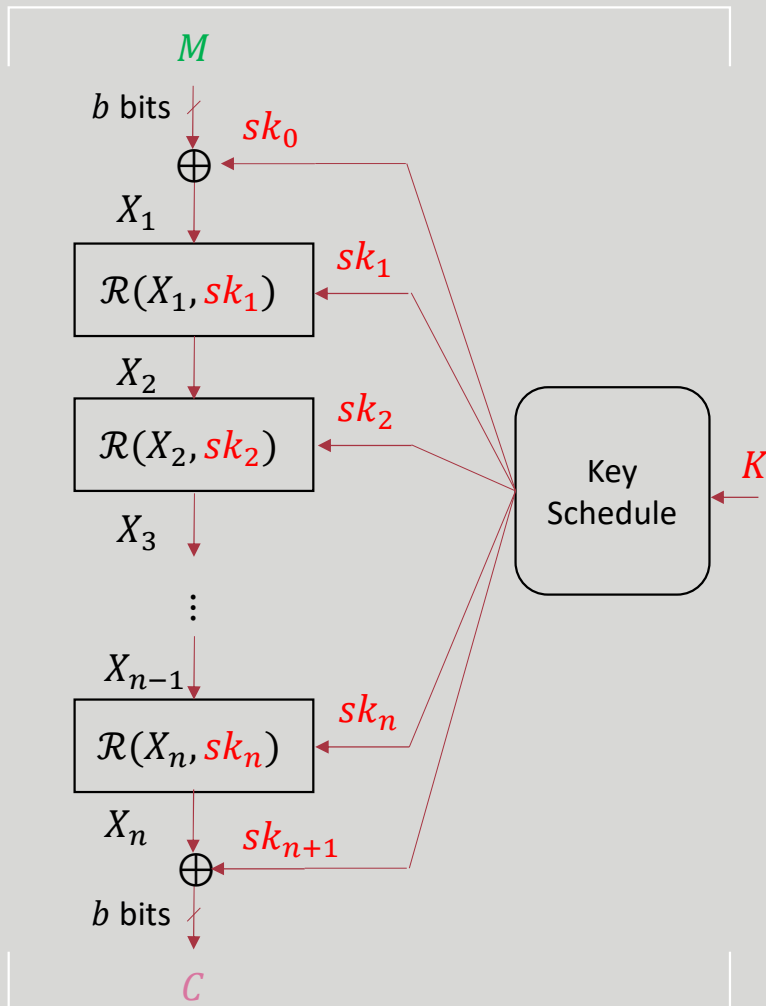
رمزنگاری

پاییز سال ۱۴۰۰



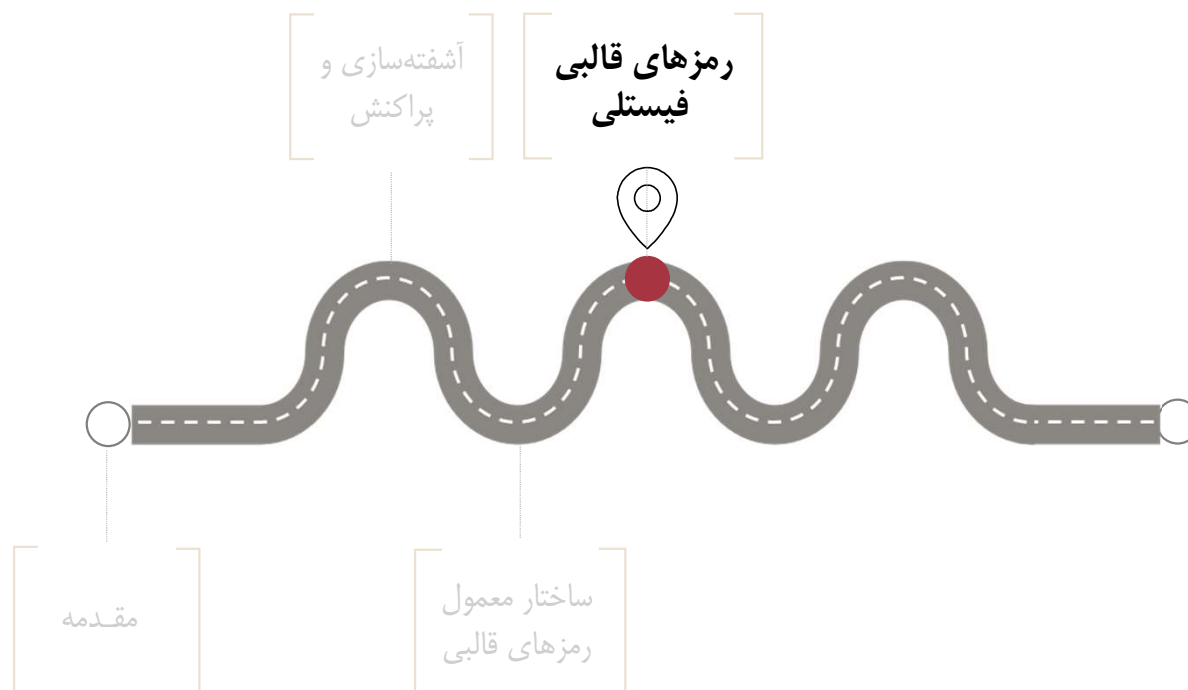
## ■ رمزهای قالبی تکرارپذیر

- از روی **کلید اصلی** ( $K$ ) توسط طرح تولید **زیرکلیدها** ( $Key$  Schedule)، بر هر دور یک زیرکلید ( $Subkey$ ) ساخته می‌شود.
- در دور  $r$ ام، مقدار  $b$  بیتی ورودی ( $X_r$ ) با استفاده از تابع دور و **زیرکلید** دور به یک مقدار  $b$  بیتی ( $X_{r+1}$ ) تبدیل می‌شود.
- بعضا در برخی از رمزهای قالبی در ابتدا و (یا) انتهای الگوریتم نیز یک **کلید** اضافه می‌شود که به **کلید سفیدسازی** (**Whitening Key**) معروف است.



## ■ دسته‌بندی فیستلی و جانشینی-جایگشتی

- ساختار تابع دور در بسیار از رمزهای قالبی به صورت شبکه‌ی فیستلی و یا شبکه‌ی جانشینی-جایگشتی (SPN) است.
- هر یک از این دو ساختار ویژگی‌ها و کاربردهای مخصوص به خود را دارند.
- به عنوان مثال اولین رمز قالبی استاندارد ارائه شده (DES) از ساختار فیستل استفاده می‌کرد.
- و یا رمز قالبی شناخته شده‌ای مانند AES که از ساختار جانشینی-جایگشتی استفاده می‌کند.
- در بخش‌های پیش رو به معرفی بیشتر این دو ساختار می‌پردازیم.



## ■ ساختار فیستل

(Feistel)

- ساختار فیستل اولین بار توسط یکی از محققین IBM به نام Horst Feistel ارائه شد.

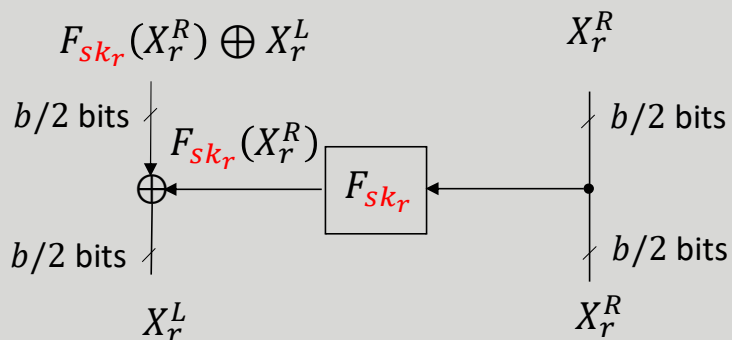
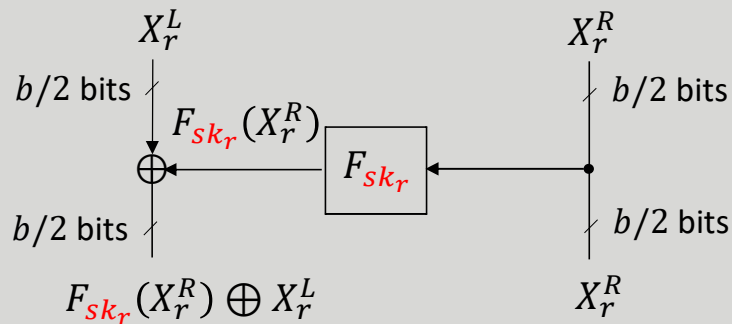
1. حالت (State) ورودی دور  $(X_r)$  به دو بخش مساوی  $X_r^R$  و  $X_r^L$  تقسیم می شود.

2. نیمی از آن  $(X_r^R)$  از تابع دور  $(F)$  عبور کرده و با نیمه دیگر XOR می شود.

3.  $X_r^R$  بدون تغییر به خروجی منتقل می شود.

- یک دور فیستلی خود وارون است.

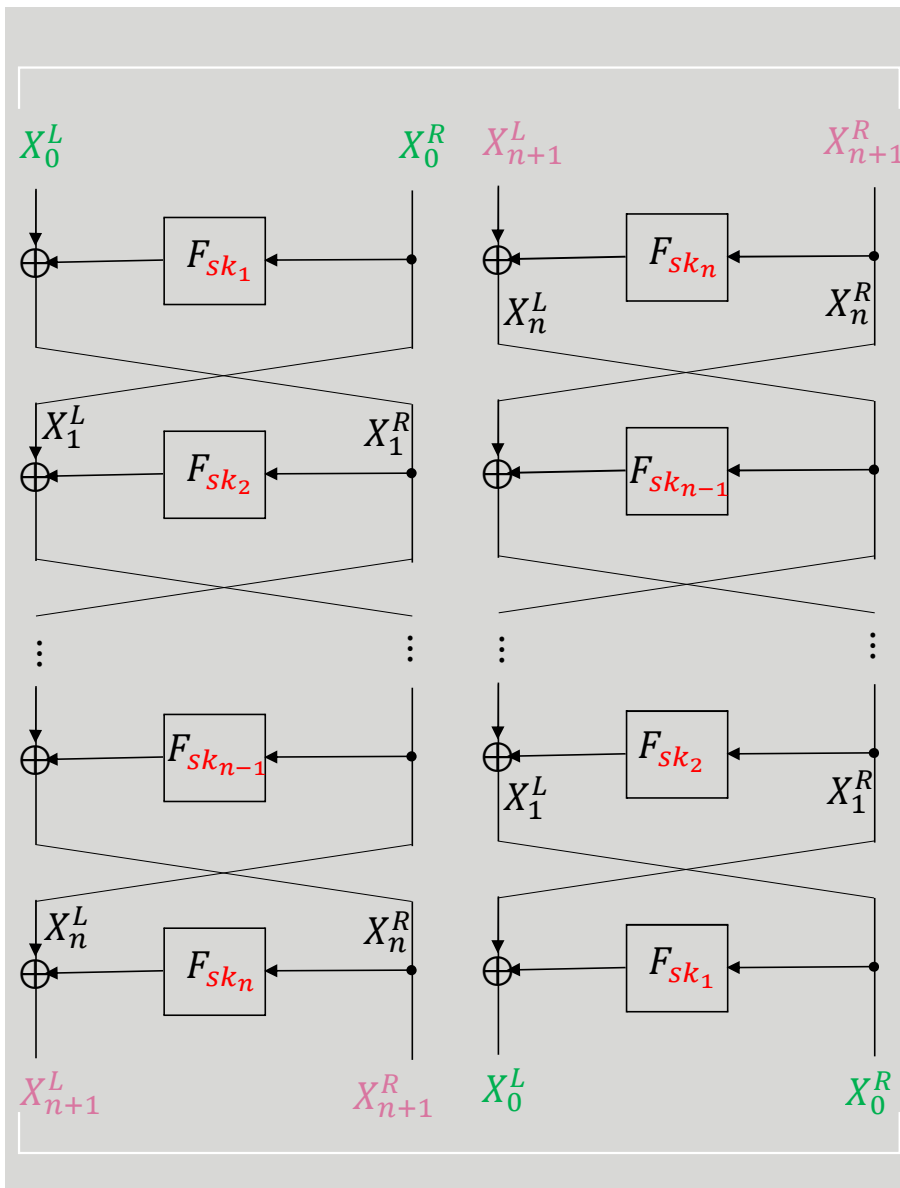
- مستقل از تابع  $F$  (حتی اگر یک به یک نباشد)، عملیات دور یک به یک است.

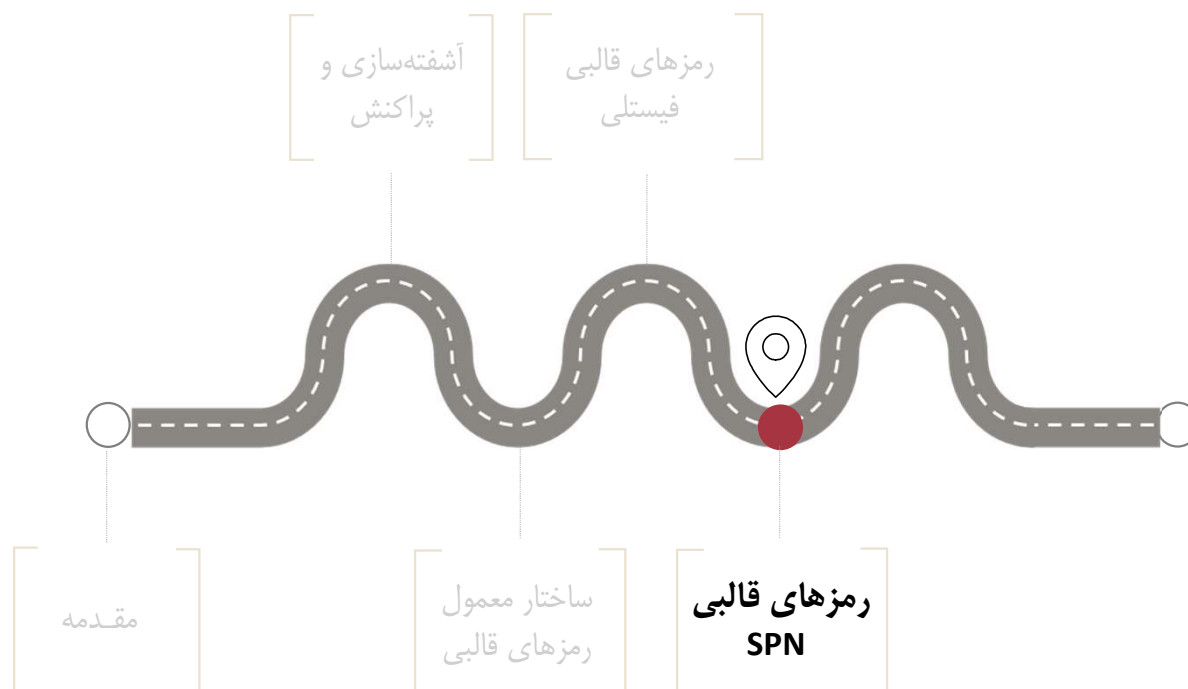


## ■ ساختار فیستل

... ادامه

- با استفاده از ساختار فیستلی، می‌توان الگوریتم‌های رمز قالبی با تعداد دوره‌های بیشتری طراحی کرد.
- به جز دور آخر، هر دور یک عمل جابه‌جایی (Swapping) است، که جای بخش‌های چپ و راست حالت را عوض می‌کند.
- در چنین ساختاری رمزگشایی با استفاده از همان الگوریتم رمزگذاری و صرفاً با تغییر ترتیب **زیر کلیدها** انجام می‌شود.

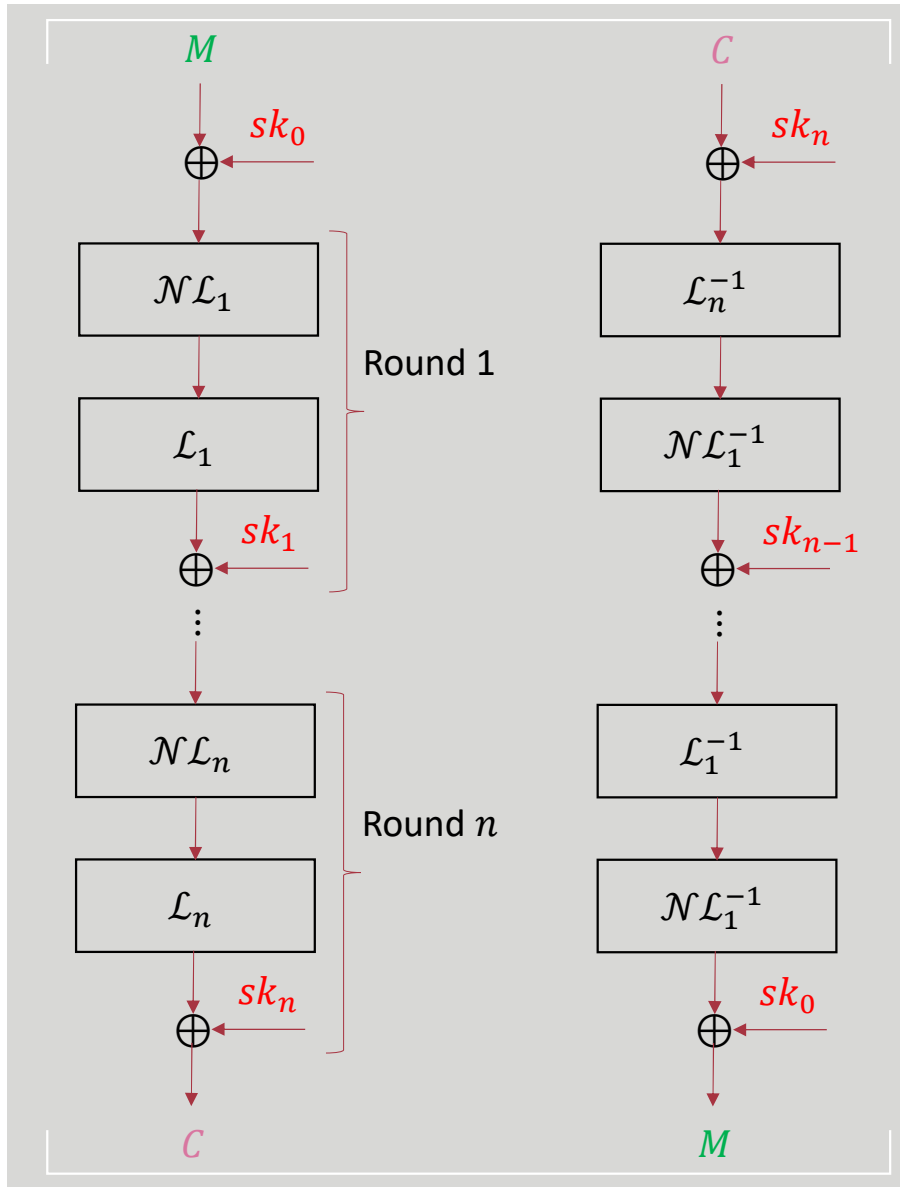




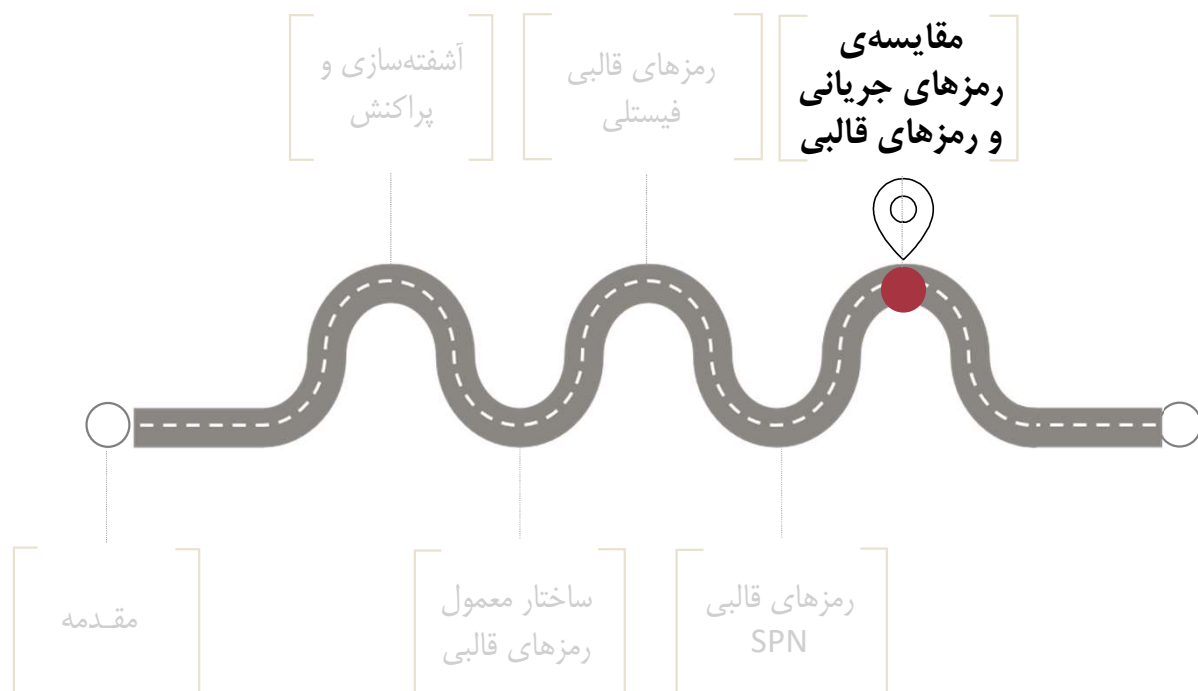
## شبکه‌ی جانشانی - جایگشتی

### (Substitution-Permutation Network)

- **کلید سفیدسازی** در ابتدای الگوریتم به **متن اصلی** اضافه می‌شود.
- هر دور شامل سه بخش است:
- لایه‌ی غیرخطی ( $\mathcal{NL}$ ) که معمولاً از توابع غیرخطی کوچک به نام جعبه‌های جانشانی (Sbox) تشکیل می‌شوند.
- لایه‌ی خطی ( $\mathcal{L}$ ) که می‌تواند شامل چند تابع مختلف خطی باشد.
- **زیرکلید** با استفاده از عملگر XOR در انتهای هر دور اضافه می‌شود.
- توابع به کار رفته در دور باید یک‌به‌یک باشند تا عملیات رمزگشایی نیز امکان‌پذیر باشد.
- معمولاً لایه‌های خطی و غیرخطی به کاررفته در دورهای مختلف به صورت یکسان یا بسیار شبیه به هم انتخاب می‌شوند.







## ■ مقایسه‌ی رمزهای جریانی و رمزهای قالبی

### مزیت‌های رمزهای جریانی در مقایسه رمزهای قالبی

- رمزهای جریانی در مقایسه با رمزهای قالبی عموماً به مساحت پیاده‌سازی کمتری نیاز داشته و می‌توانند به سرعت (بسیار) بالاتری دست یابند.
- از آنجایی که طول **متن اصلی** در رمزهای جریانی متغیر است، استفاده از آن‌ها در پروتکل‌های امنیتی راحت‌تر است.

### مزیت‌های رمزهای قالبی در مقایسه با رمزهای جریانی

- با به‌کارگیری رمزهای قالبی به عنوان یک اولیه‌ی رمزنگاری، می‌توان سایر اولیه‌های رمزنگاری (مانند MAC، رمز جریانی و ...) را نیز ساخت.
- امنیت رمزهای قالبی بهتر از رمزهای جریانی توسط جامعه‌ی رمزنگاری فهمیده می‌شود و بنابراین بیشتر مورد اعتماد هستند.

## ■ مقایسه‌ی رمزهای جریانی و رمزهای قالبی

### کاربردها

- در تعداد قابل توجهی از کاربردهای مهم، رمزهای قالبی به مرور زمان جایگزین رمزهای جریانی شده‌اند.
- در حالی که نسل دوم تلفن همراه از رمز جریانی A5/1 استفاده می‌کرد، نسل سوم از رمز قالبی Kasumi استفاده می‌کند.
- در حالی که استاندارد قبلی شبکه‌های بی‌سیم WiFi (802.11.a) از رمز جریانی RC4 استفاده می‌کرد، استاندارد جدید (802.11i) از رمز قالبی AES استفاده می‌کند.
- دو دلیل عمده برای توجه بیشتر به رمزهای قالبی:
  1. چالش‌های امنیتی رمزهای جریانی
  2. کم‌رنگ شدن مزایای رمزهای جریانی در برخی کاربردهای عملی

## ■ مقایسه‌ی رمزهای جریانی و رمزهای قالبی

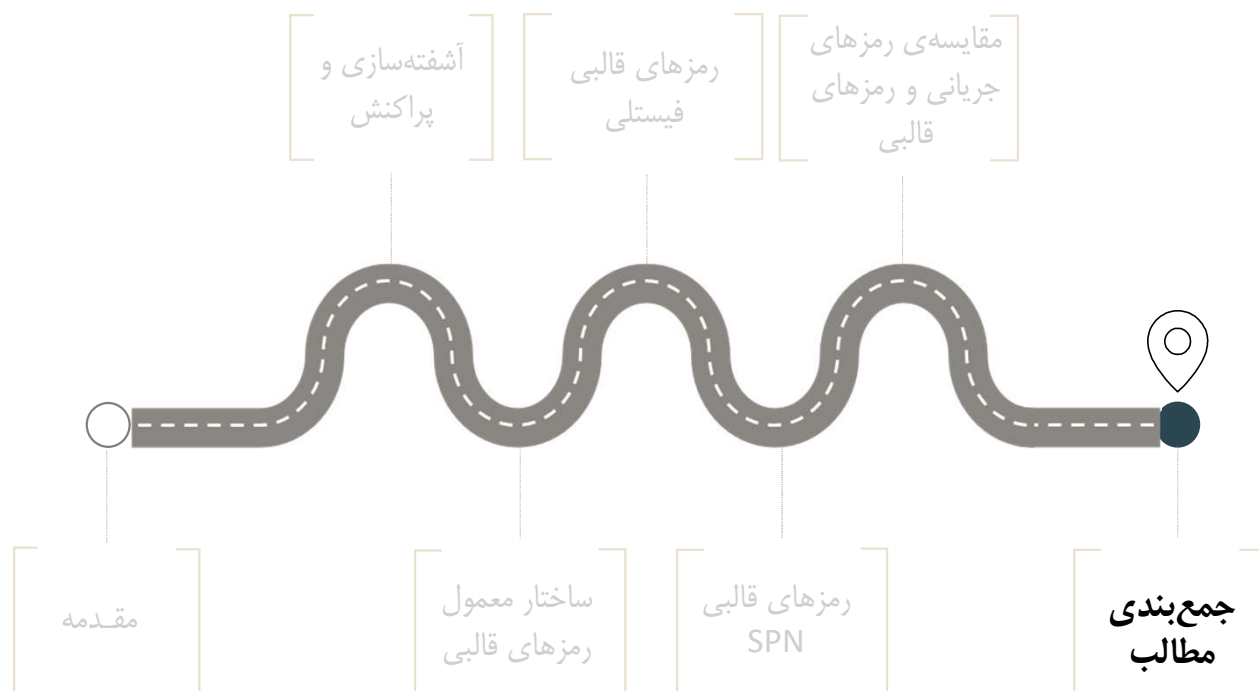
### پروژه‌ی NESSIE

- (با الهام از پروژه‌ی AES) پروژه‌ی NESSIE از سال ۱۹۹۹ تا ۲۰۰۳ توسط اتحادیه‌ی اروپا برگزار شد.
- هدف از این پروژه، طراحی اولیه‌های امن و کارای رمزنگاری بود.
- در پایان پروژه، هیچ‌کدام از کاندیداهای رمز جریانی به علت نداشتن امنیت کافی، انتخاب نشدند!

## ■ کم‌رنگ شدن مزایای رمزهای جریانی

- پیشرفت‌های سریع در پیاده‌سازی مدارهای مجتمع سبب شده است که پیاده‌سازی‌های بزرگ‌تر به راحتی امکان‌پذیر شده و مساحت پیاده‌سازی تاثیر کم‌تری بر روی قیمت داشته باشد.
- رمزهای قالبی مدرن (نظیر AES) سرعت کافی برای بیشتر کاربردهای مهم را دارند و در عمل به سرعت‌های خیلی بالاتر از آن نیاز چندانی وجود ندارد.
- بنابراین در شرایطی که طراحی رمزهای قالبی راحت‌تر و امنیت آن‌ها قابل فهم‌تر است، تمایل به استفاده از رمزهای جریانی کم‌تر می‌شود.

- البته مطالب گفته شده به معنای حذف کامل رمزهای جریانی نیست!
- در نوامبر ۲۰۰۴ توسط یک پروژه‌ی از اتحادیه اروپا (ECRYPT) فراخوانی به منظور طراحی رمزهای جریانی اعلام شد.
- هدف انتخاب رمزهای جریانی در قالب مسابقه‌ای به نام eSTREAM بود.
- مسابقه در آپریل ۲۰۰۸ به اتمام رسید.
- ۴ الگوریتم با نام‌های HC-128, Rabbit, Salsa20/12 و SOSEMANUK برای کاربردهای نرم‌افزاری با سرعت بسیار بالا انتخاب شدند.
- ۳ الگوریتم با نام‌های Grain, MICKEY و Trivium برای پیاده‌سازی سخت‌افزاری با محدودیت زیاد انتخاب شدند.
- برخی از این الگوریتم‌ها نظیر Grain در صنعت مورد استفاده قرار گرفته است.



## ■ جمع‌بندی مطالب



- امروزه مزایای رمزهای جریانی نسبت به رمزهای قالبی با توجه به نیازهایی که در کاربردهای عملی وجود دارند، کم‌رنگ‌تر شده‌اند.
- رمزهای قالبی استاندارد (به خصوص AES) در طیف وسیعی از کاربردها استفاده می‌شوند.
- رمزهای جریانی نیز در برخی از کاربردهای خاص استفاده می‌شوند.
- به عنوان مثال، رمز جریانی ZUC که در استاندارد شبکه‌های موبایل LTE استفاده می‌شود.