Shahid Beheshti University

رمزنگاری

هادی سلیمانی

پژوهشکده فضای مجازی دانشگاه شهید بهشتی

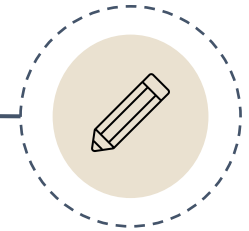
مجوز استفاده و نشر

- 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/

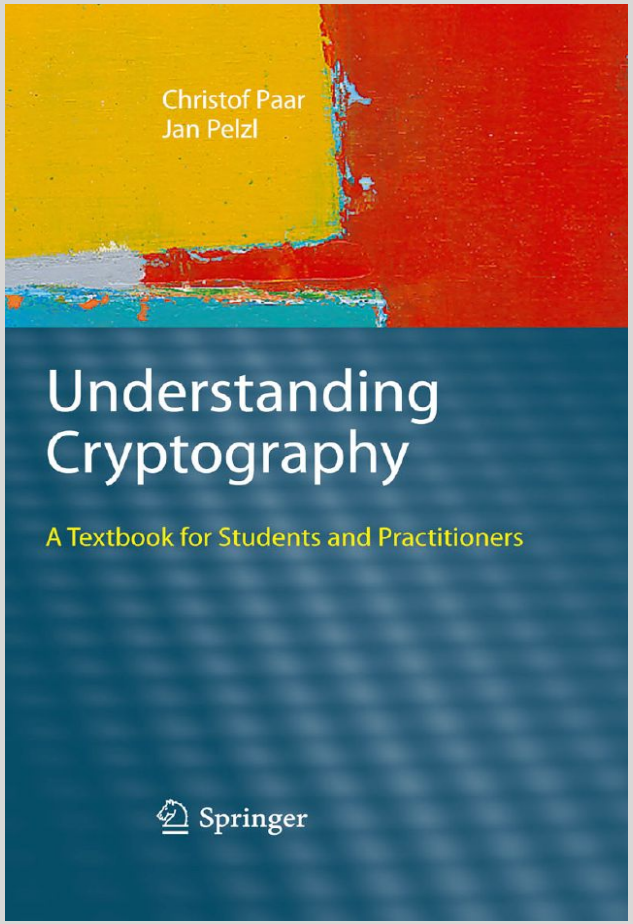
درس هفتم



الگوریتم رمزنگاری AES

معرفی مـرجع

الگوریتم رمزنگاری AES



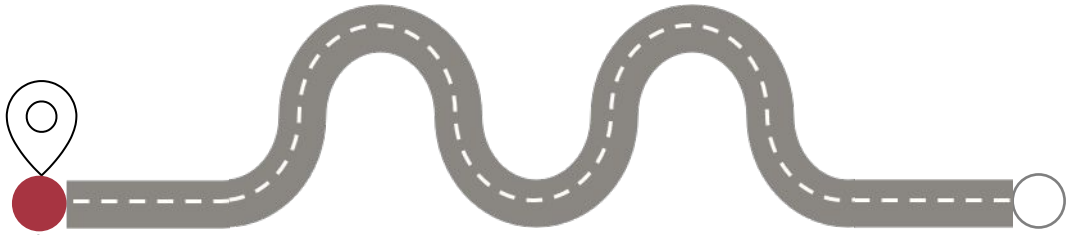
Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.

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

فهرست عناوین درس

الگوریتم رمزنگاری AES

- مقدمه
- توصیف الگوریتم AES
- انتخاب اجزاء AES
- پیاده‌سازی نرم‌افزاری AES
- جمع‌بندی مطالب



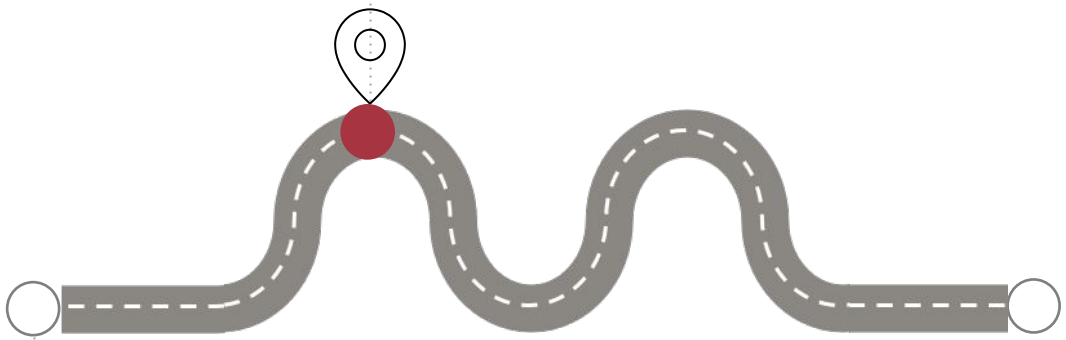
مقدمه

AES

(Advanced Encryption Standard)

- در سال 1997 مسابقه‌ی AES به منظور انتخاب الگوریتم رمزنگاری استاندارد توسط NIST برگزار شد.
- 15 طرح در این مسابقه شرکت کردند.
- در اکتبر 2000، الگوریتم Rijndael به عنوان برنده اعلام و در نوامبر 2001 رسماً به عنوان استاندارد معرفی شد.
- از آن زمان به بعد، این الگوریتم AES نام‌گذاری شد.
- طراحان AES دو رمزنگاری بلژیکی به نام‌های Vincent Rijmen و Joan Daemen هستند.

توصیف
الگوریتم AES

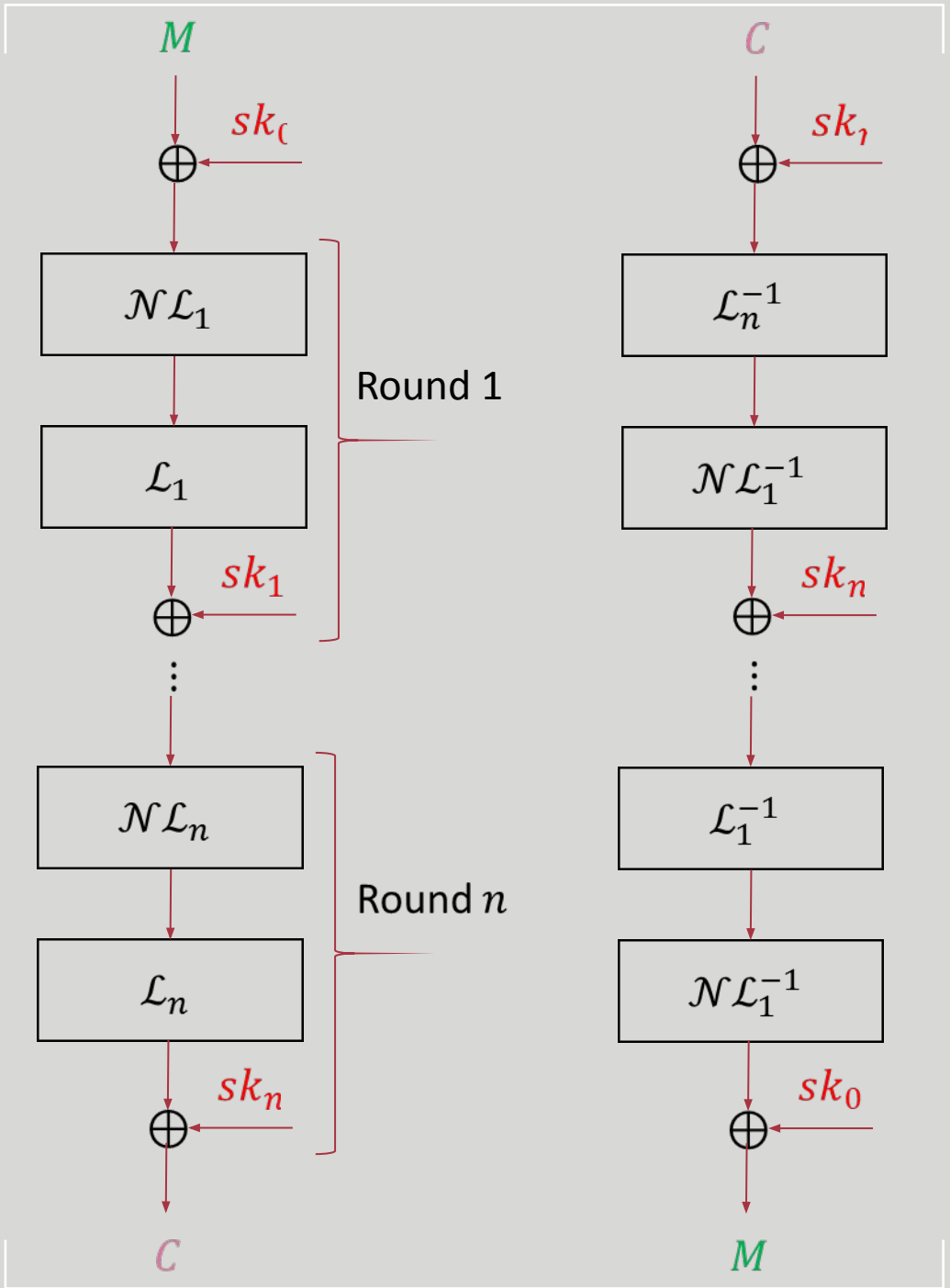


مقدمه

شبکه‌ی جانشانی ـ جایگشتی

یادآوری

- 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 می‌شود.
  - لایه‌ی خطی ($\mathcal{L}$) که می‌تواند شامل چند تابع خطی مختلف باشد.
  - زیرکلید با استفاده از عملگر Xor در انتهای هر دور اضافه می‌شود.
- توابع به‌کاررفته در دور باید یک‌به‌یک باشند تا عملیات رمزگشایی نیز امکان‌پذیر باشد.
- معمولا لایه‌های خطی و غیرخطی به‌کاررفته در دورهای مختلف به‌صورت یکسان یا بسیار شبیه به هم انتخاب می‌شوند.

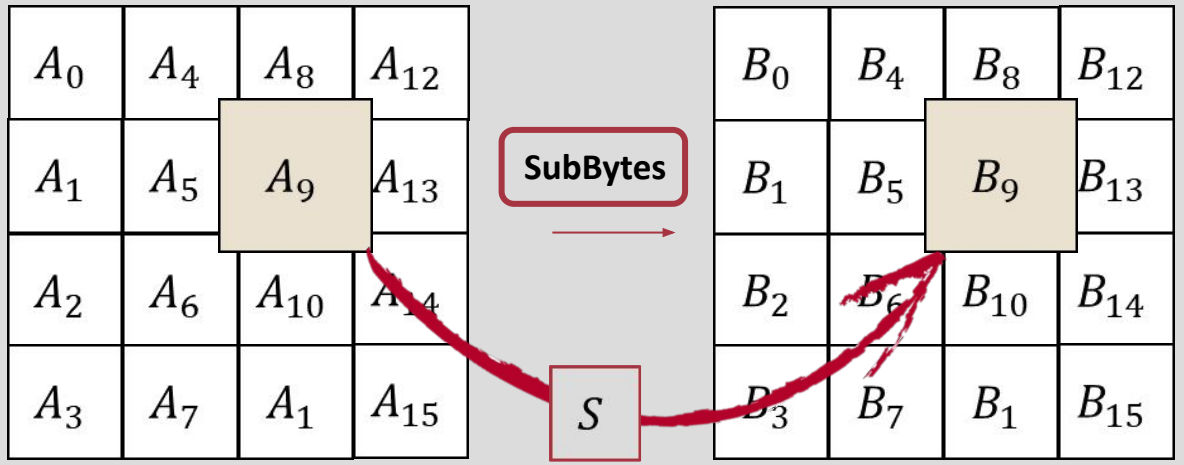


ساختار کلی AES

- AES یک رمز قالبی مبتنی بر ساختار SPN و طول قالب 128 بیت (16 بایت) است.
- طول کلید می‌تواند یکی از مقادیر 128، 192 و 256 بیت باشد.
- در سه نسخه‌ی متفاوت که براساس طول کلید به ترتیب با AES-192 ،AES-128 و AES-256 نمایش داده می‌شوند، تعداد دورها به ترتیب 10، 12 و 14 است.
- هر دور (به جز دور آخر) شامل چهار گام است:
  1. گام جايگذاري بايتي
  2. گام شيفت سطري
  3. گام مخلوطساز ستوني (به جز دور آخر)
  4. گام جمع با زیرکلید
- در ابتداي الگوريتم، کلید سفیدسازی اضافه می‌شود.

گام جای‌گذاری بایتی

- اگر ۱۶ بایت ورودی دور را با $(A_0, A_1, \ldots, A_{15})$ نمایش دهیم، براساس توصیف جعبه‌ی جانشانی AES هر بایت A_i با یک بایت دیگر B_i جایگزین می‌شود.
- برخلاف DES، مشخص است که جعبه‌ی جانشانی AES $(S: A \to B)$ چگونه و براساس چه معیارهایی انتخاب شده است (در درس رمزنگاری پیشرفته بیشتر به آن می‌پردازیم).



This picture is inspired by the picture given in Wikipedia.

گام جای‌گذاری بایتی

جعبه‌ی جانشانی AES

۱. ابتدا معکوس ورودی (به جز مقدار 0) در میدان زیر محاسبه می‌شود:

$$GF(2^8) = GF(2)[/(x^8 + x^4 + x^3 + x + 1)$$

- معکوس 0 را خود 0 تعریف می‌کنیم.
- محاسبه‌ی معکوس با استفاده از الگوریتم توسعه‌یافته‌ی اقلیدسی انجام می‌شود.
- مثال:

$$A_i = (C2)_{hex} = (1100\ 0010)_2 = \mathrm{x}^7 + \mathrm{x}^6 + x$$
$$B_i' = A_i^{-1} = x^5 + x^3 + x^2 + x + 1 = (0010\ 1111)_2$$

۲. پس از آن، تبدیل آفینی روبه‌رو در $GF(2)$ اعمال می‌شود:

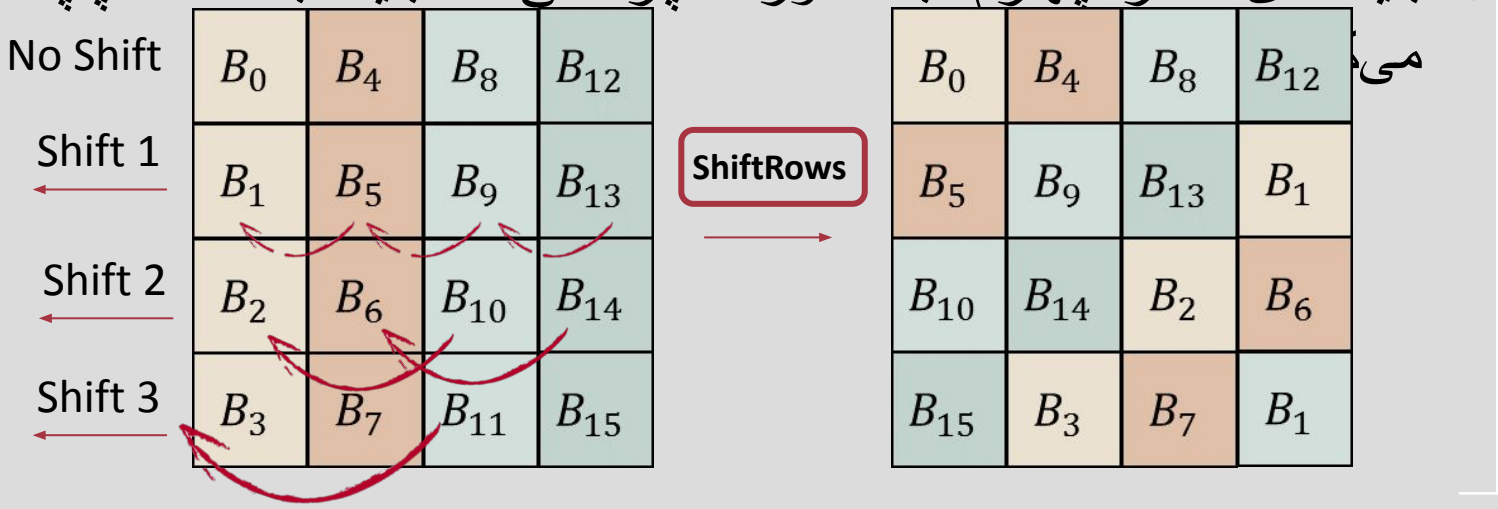
- در پیاده‌سازی (در صورت وجود حافظه کافی) می‌توان جعبه‌ی جانشانی را پیش‌محاسبه و به‌صورت یک جدول ذخیره کرد.

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

$$B_i = (0010\ 0101)_2 = (25)_{hex}$$

گام شیفت سطری

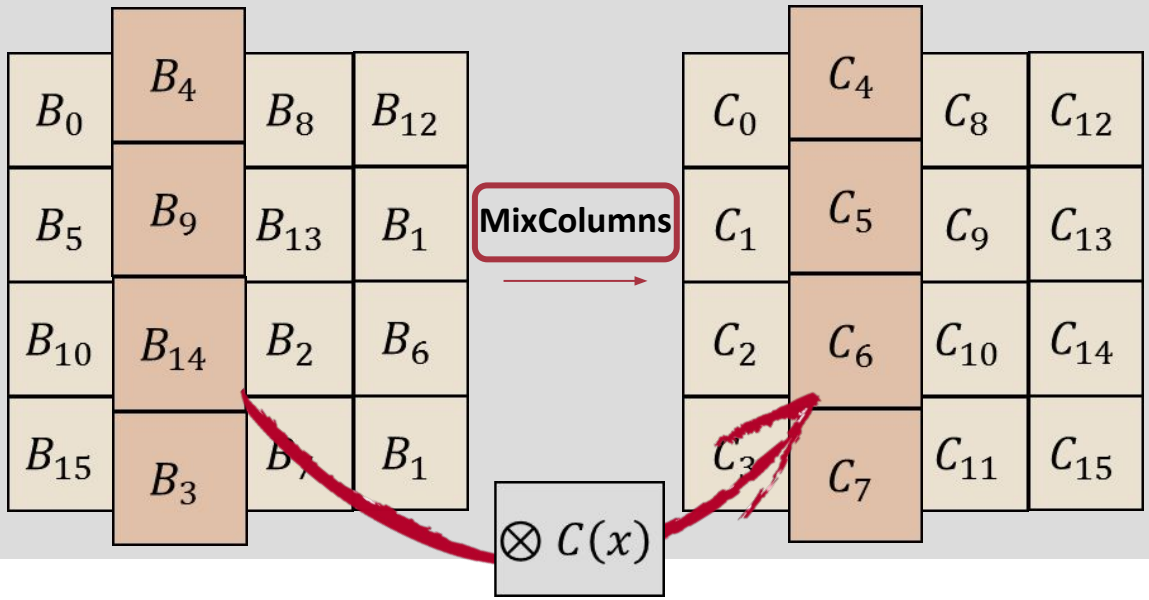
- سطر اول تغییر نمی‌کند.
- بایت‌های سطر دوم، به صورت چرخشی یک بایت به سمت چپ شیفت پیدا می‌کنند.
- بایت‌های سطر سوم، به صورت چرخشی دو بایت به سمت چپ شیفت پیدا می‌کنند.
- بایت‌های سطر چهارم، به صورت چرخشی سه بایت به سمت چپ شیفت پیدا می‌



This picture is inspired by the picture given in Wikipedia.

گام مخلوط سازي ستوني (به جز دور آخر)

- هر چهار بایت واقع در یک ستون با ضرب در یک ماتریس به چهار بایت دیگر تبدیل می‌شوند.
- این تابع خطی و یک‌به‌یک است.



This picture is inspired by the picture given in Wikipedia.

گام مخلوط سازي ستوني (به جز دور آخر)

... ادامه

- تمامی محاسبات ضرایب در میدان زیر انجام می‌شوند:

$$GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$$

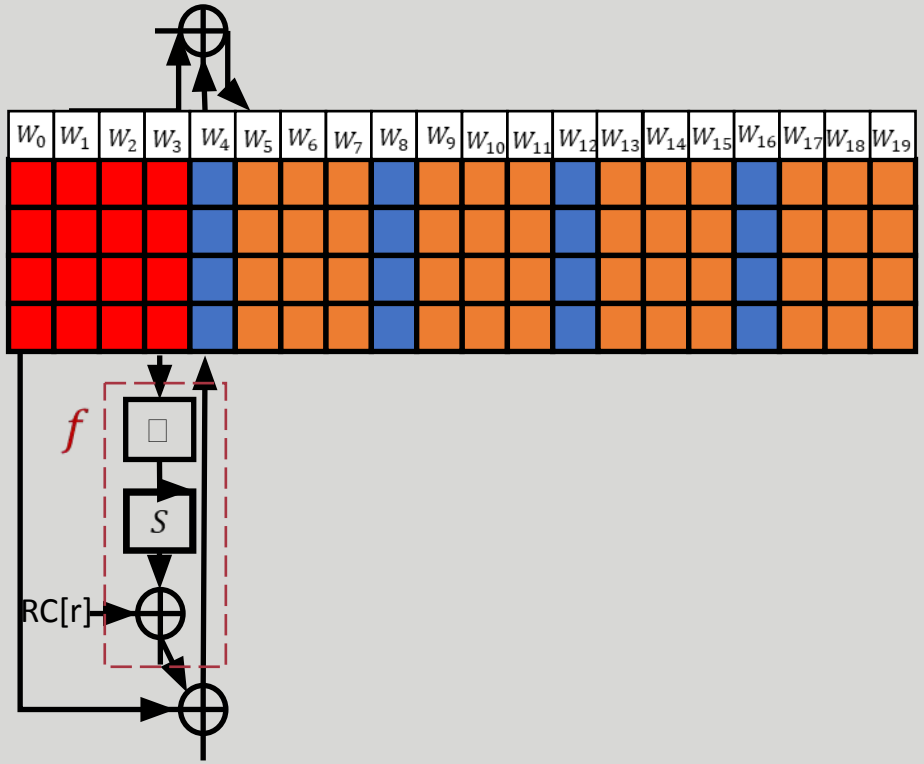
- مثال:

$$02.25 = x.(x^5 + x^2 + 1) = x^6 + x^3 + x$$

$$03.25 = (x + 1).(x^5 + x^2 + 1)$$
$$= (x^6 + x^3 + x\ ) + (x^5 + x^2 + 1)$$
$$= x^6 + x^5 + x^3 + x^2 + x + 1$$

$$\begin{bmatrix} C_4 \\ C_5 \\ C_7 \\ C_8 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} B_4 \\ B_9 \\ B_{14} \\ B_3 \end{bmatrix}$$

طرح تولید زیرکلید برای AES-128



- ۱۲۸ بیت کلید اصلی را معادل چهار کلمه ۳۲ بیتی در نظر می‌گیریم و با w_0، w_1، w_2 و w_3 نشان می‌دهیم.
- کلمه‌ی w_i بدین صورت تولید می‌شود:

$$w_i = f(w_{i-1}) \oplus w_{i-4} \quad \text{if } i \bmod 4 = 0$$
$$w_i = w_{i-1} \oplus w_{i-4} \quad \text{if } i \bmod 4 \neq 0$$

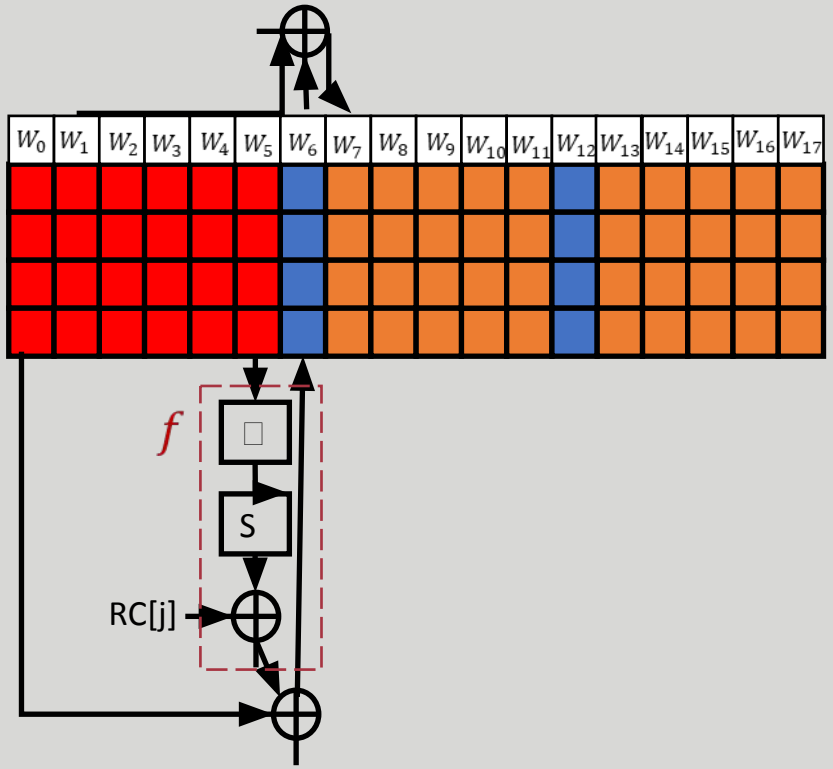
- $RC[r]$ در تابع f، یک مقدار ثابت ۸ بیتی است که تنها به اولین بایت سمت چپ اضافه شده و در هر دور مقداری متفاوت با دورهای دیگر دارد.
- ۴ کلمه‌ی $(w_{4r}, w_{4r+1}, w_{4r+2}, w_{4r+3})$ زیرکلید rام را تشکیل می‌دهند.

طرح تولید زیرکلید برای AES-192

- ۱۹۲ بیت کلید اصلی را معادل شش کلمه ۳۲ بیتی در نظر می‌گیریم و با w_0، w_1، w_2، w_3، w_4 و w_5 نشان می‌دهیم.
- کلمه‌ی w_i بدین صورت تولید می‌شود:

$$w_i = f(w_{i-1}) \oplus w_{i-6} \quad \text{if } i \bmod 6 = 0$$
$$w_i = w_{i-1} \oplus w_{i-6} \quad \text{if } i \bmod 6 \neq 0$$

- تابع f و فرمول محاسبه‌ی مقدار ثابت $RC[j]$ همانند طرح تولید زیرکلید AES-128 تعریف شده است.
- ۴ کلمه‌ی $(w_{4r}, w_{4r+1}, w_{4r+2}, w_{4r+3})$ زیرکلید دور rام را تشکیل می‌دهند.

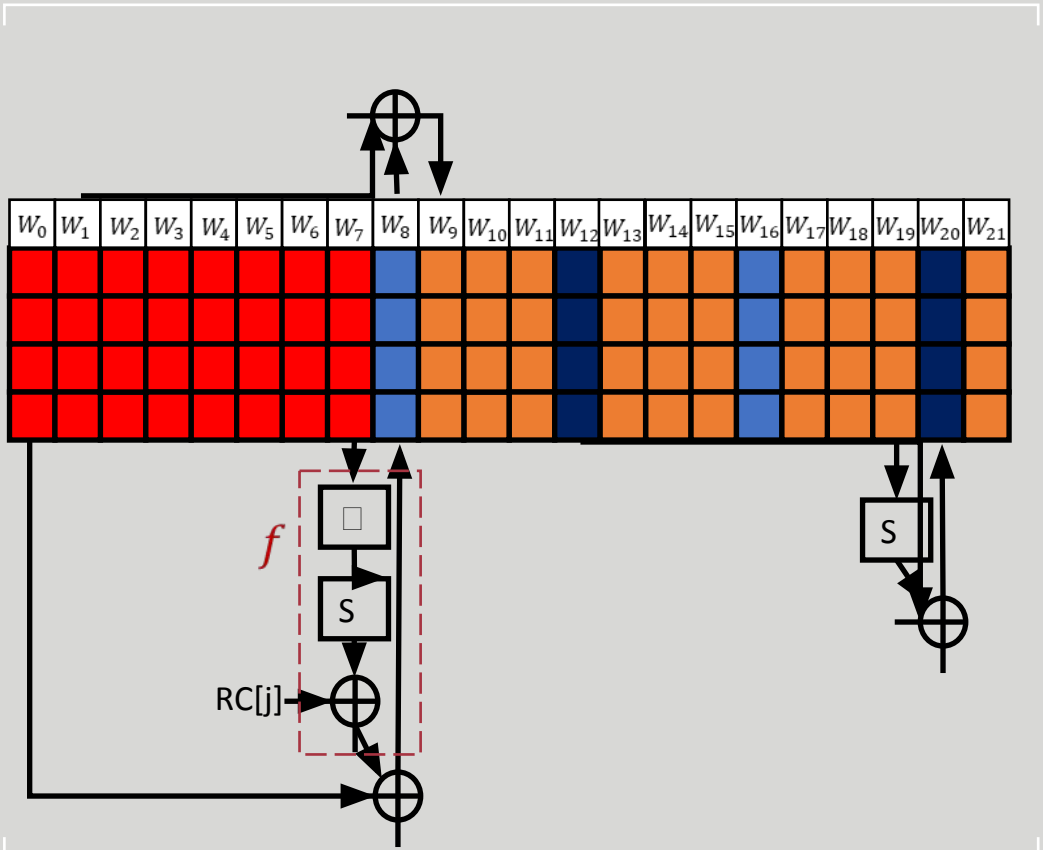


طرح تولید زیرکلید برای AES-256

- ۲۵۶ بیت کلید را معادل هشت کلمه ۳۲ بیتی در نظر می‌گیریم و با w_0، w_1، w_2، w_3، w_4، w_5، w_6 و w_7 در نظر می‌گیریم.
- کلمه‌ی w_i بدین صورت تولید می‌شود:

$$w_i = f(w_{i-1}) \oplus w_{i-8} \text{ if } i \equiv 0 \bmod 8$$
$$w_i = w_{i-1} \oplus w_{i-8} \text{ if } i \not\equiv 0 \bmod 4$$
$$w_i = S(w_{i-1}) \oplus w_{i-8} \text{ if } i \equiv 4 \bmod 8$$

- تابع f و فرمول محاسبه‌ی مقدار ثابت $RC[j]$ همانند طرح تولید زیرکلید AES-128 تعریف شده است.
- ۴ کلمه‌ی $(w_{4r}, w_{4r+1}, w_{4r+2}, w_{4r+3})$ زیرکلید دور rام را تشکیل می‌دهند.



مقادیر ثابت در طرح تولید زیرکلید AES

- مقدار $RC[i]$ در میدان $GF(2^8) = GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$ به‌صورت زیر محاسبه می‌شود:

$$RC[i] = x^{i-1}$$

- مثال:

$$RC[3] = x^2$$
$$RC[9] = x^8 = x^4 + x^3 + x + 1$$

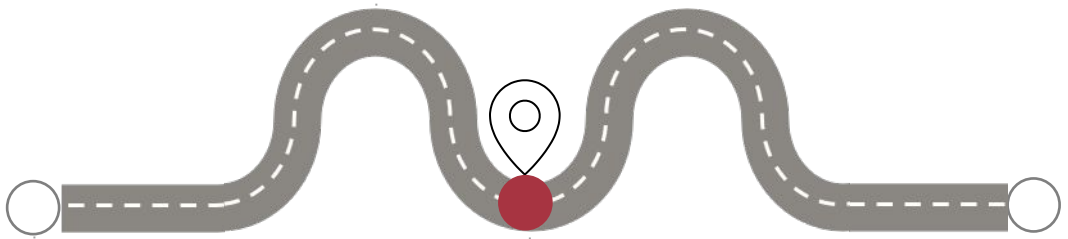
پیاده‌سازی طرح تولید زیرکلید

- طرح تولید زیرکلید AES را به دو صورت می‌توان پیاده‌سازی کرد:

1. تمامی زیرکلیدها را یک‌بار پیش‌محاسبه کرده و سپس عملیات رمزگذاری (یا رمزگشایی) را اجرا کرد.
   - این روش در حالتی‌که حافظه‌ی کافی موجود باشد مناسب است.
2. هر بار زیرکلید دورها را به‌ترتیب تولید کرد.

- از آنجایی‌که رمزگشایی از زیرکلید دور آخر شروع می‌شود، نیاز است که ابتدا این مقدار محاسبه شود.
- به همین علت سرعت رمزگشایی کمی پایین‌تر است.

توصیف
الگوریتم AES



مقدمه

انتخاب اجزاء
AES

جعبه‌ی جانشانی AES

- جعبه‌ی جانشانی به کاررفته در AES اولین بار توسط Kaisa Nyberg معرفی شده بود.
- این جعبه‌ی جانشانی در مقابل تحلیل‌های شناخته‌شده و مهم خطی و تفاضلی بیشترین مقاومت را دارد.
- جعبه‌ی جانشانی در مقابل تحلیل‌های جبری نیز مقاومت مناسبی دارد چراکه درجه‌ی جبری آن حداکثر است.
- تاکنون تلاش‌های متعددی برای استفاده از توصیف جبری جعبه‌ی جانشانی AES صورت گرفته است که به نتیجه‌ی خاصی منجر نشده‌اند.

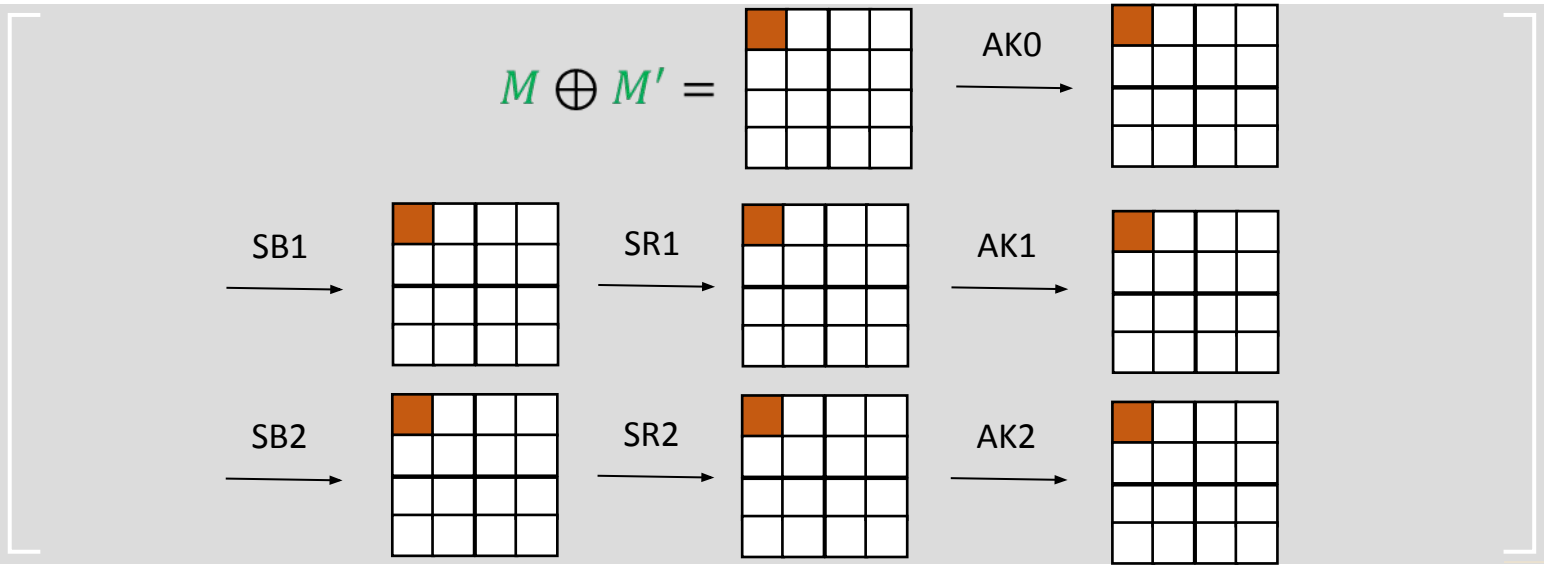
خاصیت مخلوطساز ستونی AES

- اگر چهار بایت یک ستون هم‌زمان صفر نباشند، حداقل تعداد بایت‌های غیرصفر در ورودی و خروجی برابر ۵ است.
- به عبارت دیگر اگر ورودی n بایت غیرصفر داشته باشد، آن‌گاه خروجی دارای حداقل $5 - n$ بایت غیرصفر خواهد بود.
- به طور خاص، اگر ورودی فقط یک بایت غیرصفر داشته باشد، آن‌گاه خروجی دارای ۴ بایت غیرصفر خواهد بود.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

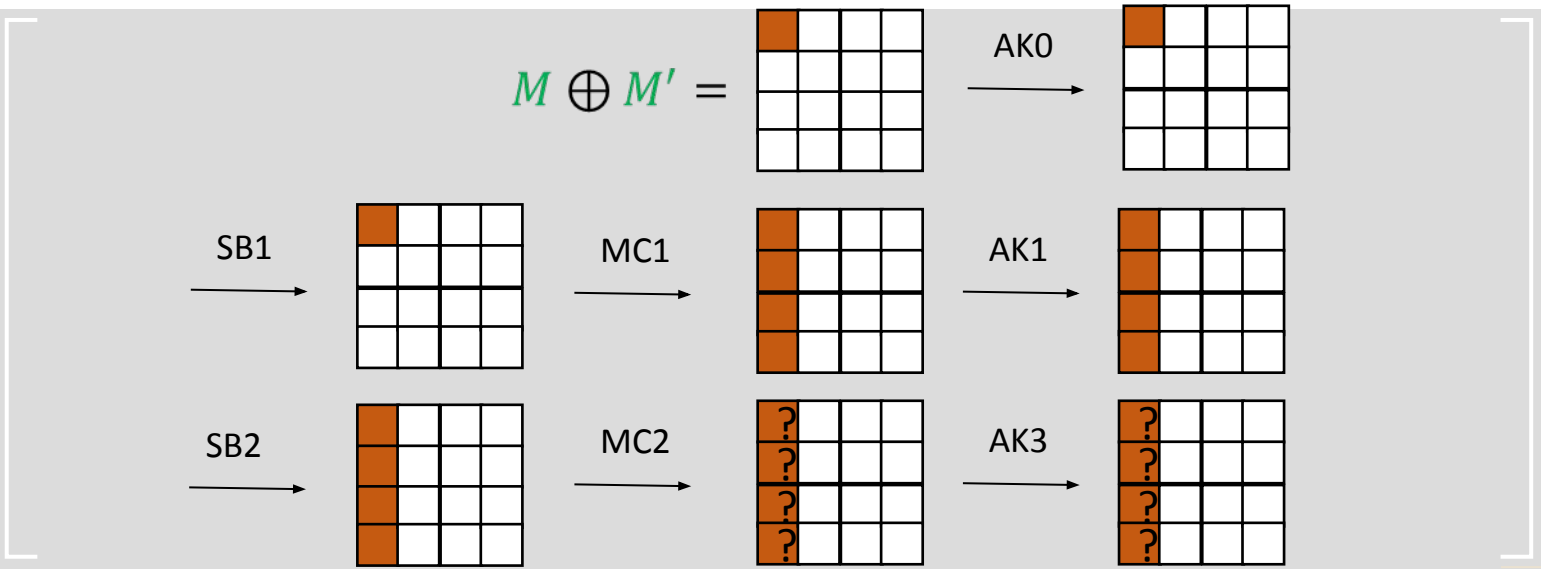
AES بدون مخلوطساز ستونی

- دو متن اصلی M و M' را در نظر می‌گیریم که فقط در بایت اول متفاوت باشند و مقادیر سایر بایت‌ها با هم برابر باشند.
- در این صورت متون رمزشده‌ی معادل آن‌ها تنها در یک بایت تفاوت دارند.
- به عبارت دیگر معیار بهمنی صادق نیست و پراکنش به‌درستی انجام نمی‌شود.



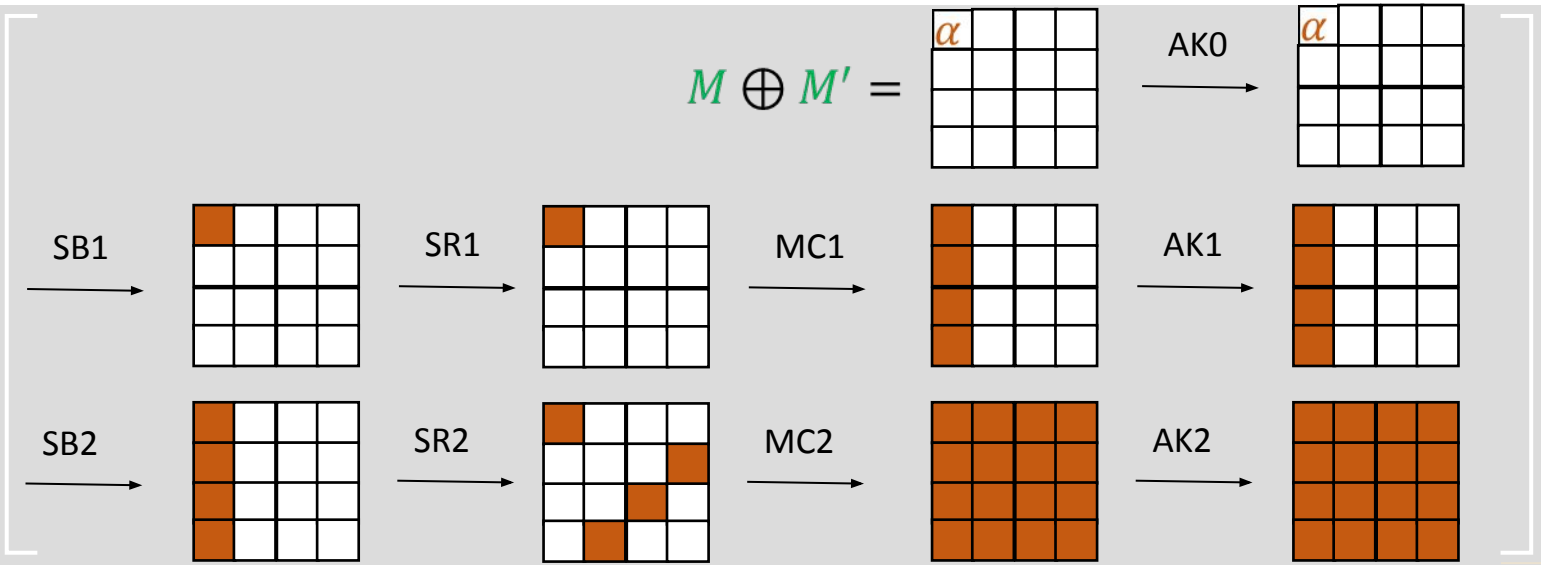
AES بدون شیفت سطری

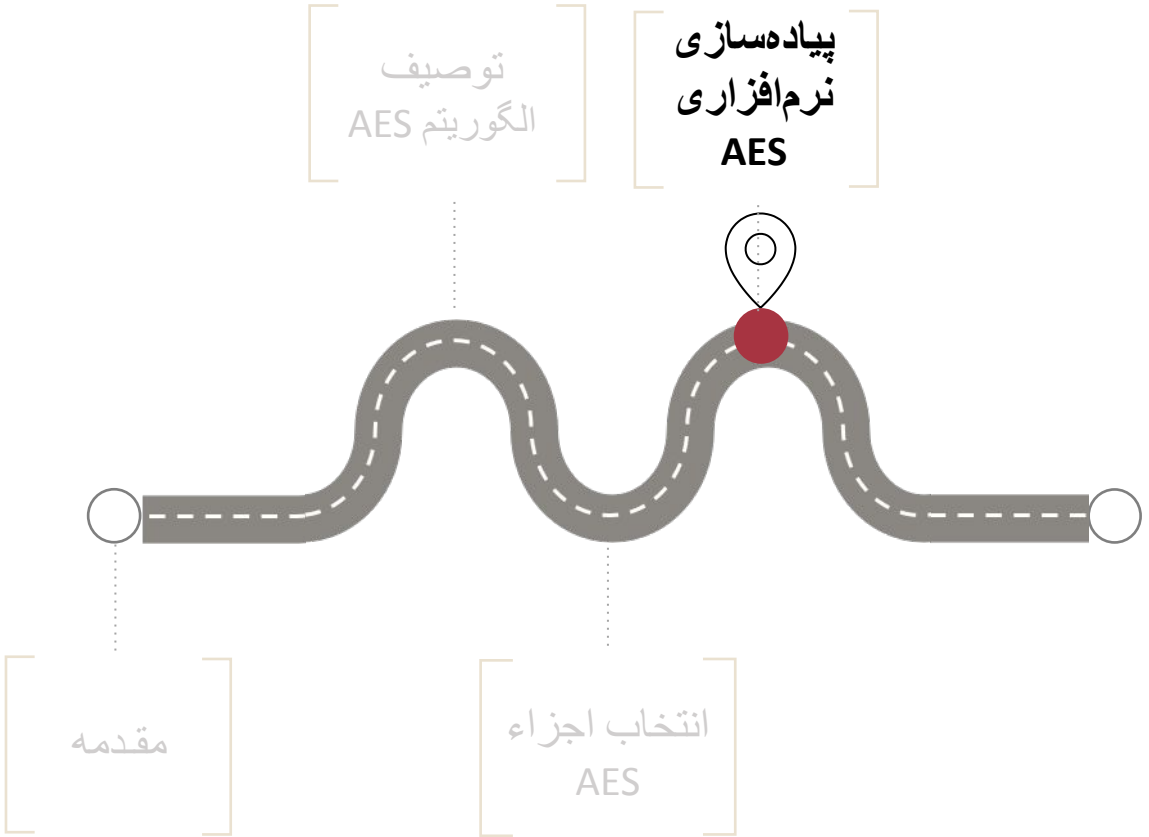
- دو متن اصلی M و M' را در نظر می‌گیریم که فقط در بایت اول متفاوت باشند و مقادیر سایر بایت‌ها با هم برابر باشند.
- در این صورت متون رمزشده‌ی معادل آن‌ها تنها در چهار بایت تفاوت دارند.
- به عبارت دیگر معیار بهمنی صادق نیست و پراکنش به‌درستی انجام نمی‌شود.



AES بدون تغییر

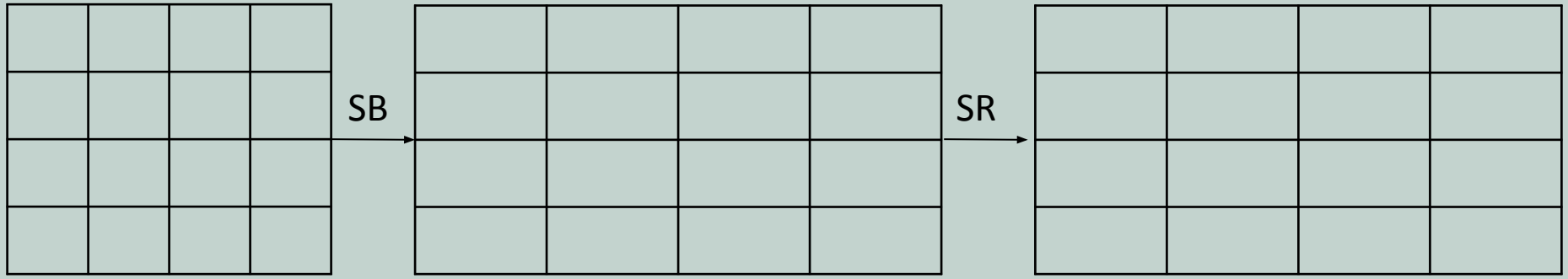
- دو متن اصلی M و M' را در نظر می‌گیریم که فقط در بایت اول متفاوت باشند و مقادیر سایر بایت‌ها با هم برابر باشند.
- اصطلاحاً گفته می‌شود که رمز پس از دو دور به پراکنش کامل (Complete Diffusion) یا (Full Diffusion) رسیده است.
- یعنی اثر تغییر یک بیت (بایت) در تمام بیت‌های (بایت‌های) خروجی دیده می‌شود.
- با افزایش تعداد دورها، معیار بهمنی برآورده می‌شود.




توصیف
الگوریتم AES
پیاده‌سازی
نرم‌افزاری
AES
مقدمه
انتخاب اجزاء
AES

یک دور (AESبدون اضافه شدن کلید)

(یادآوری)



- مخلوطساز ستونی برای ستون اول:

MC →

- نکته: مخلوطساز ستونی به سایر ستون‌ها نیز به‌صورت مشابه اعمال می‌شود.

پیش‌محاسبه‌ی تابع دور AES (به جز دور آخر)

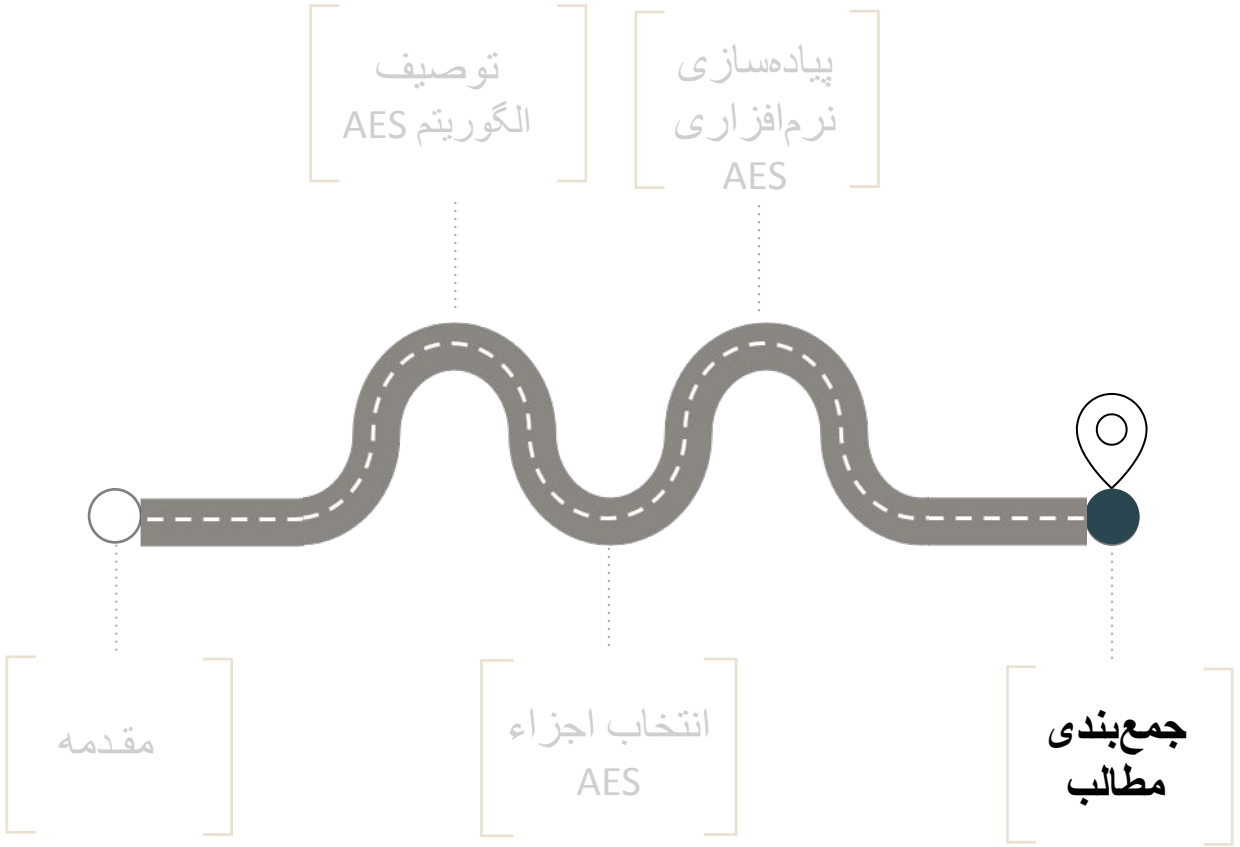
پیاده‌سازی دور آخر AES

- راهکارهای پیاده‌سازی دور آخر AES (که عملیات مخلوط‌ساز ستونی را ندارد):

1. می‌توان برای دور آخر یک جدول جداگانه ساخت.
2. می‌توان برای محاسبه‌ی مقادیر شیفت یافته‌ی $S(x_i)$ ها از جداول چهارگانه استفاده کرد (روش پیش‌محاسبه).

$$T_0[z] = \begin{bmatrix} S(z) \\ \\ \\ \end{bmatrix} \qquad T_1[z] = \begin{bmatrix} \\ \\ \\ S(z) \end{bmatrix} \qquad T_2[z] = \begin{bmatrix} S(z) \\ \\ \\ \end{bmatrix} \qquad T_3[z] = \begin{bmatrix} S(z) \\ \\ \\ \end{bmatrix}$$

- در مقالات عموما از پیاده‌سازی توصیف شده در این بخش به عنوان T-table نام برده می‌شود.
- این پیاده‌سازی اولین بار توسط طراحان AES ارائه شد.


توصیف
الگوریتم AES
پیاده‌سازی
نرم‌افزاری
AES
مقدمه
انتخاب اجزاء
AES
جمع‌بندی
مطالب

جمع‌بندی

- ویژگی خاص AES ساختار ساده و امنیت قابل‌فهم آن است.
- طراحان الگوریتم توانستند روش ساختارمندی را برای اثبات امنیت الگوریتم در مقابل تحلیل‌های مهم و شناخته‌شده‌ی تفاضلی و خطی ارائه کنند.
- استراتژی طراحان AES به‌صورت گسترده‌ای مورد استقبال سایر رمزنگاران نیز قرار گرفته است و طراحی‌های بسیاری از اصول به‌کاررفته در طراحی AES استفاده کرده‌اند.
- تاکنون ضعف خاصی برای الگوریتم رمزنگاری AES کشف نشده است.
- بهترین تحلیل ارائه شده بر روی AES که سرعتی سریع‌تر از حمله‌ی جست‌وجوی جامع داشته باشد، به ۷ دور AES-128 و ۸ دور AES-192 و ۹ دور AES-256 قابل‌اعمال است.