Shahid Beheshti University

رمزنگاری

هادی سلیمانی

پژوهشکده فضای مجازی دانشگاه شهید بهشتی

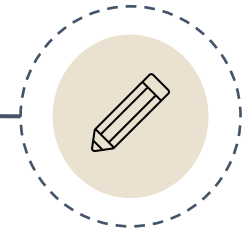
مجوز استفاده و نشر

- 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/

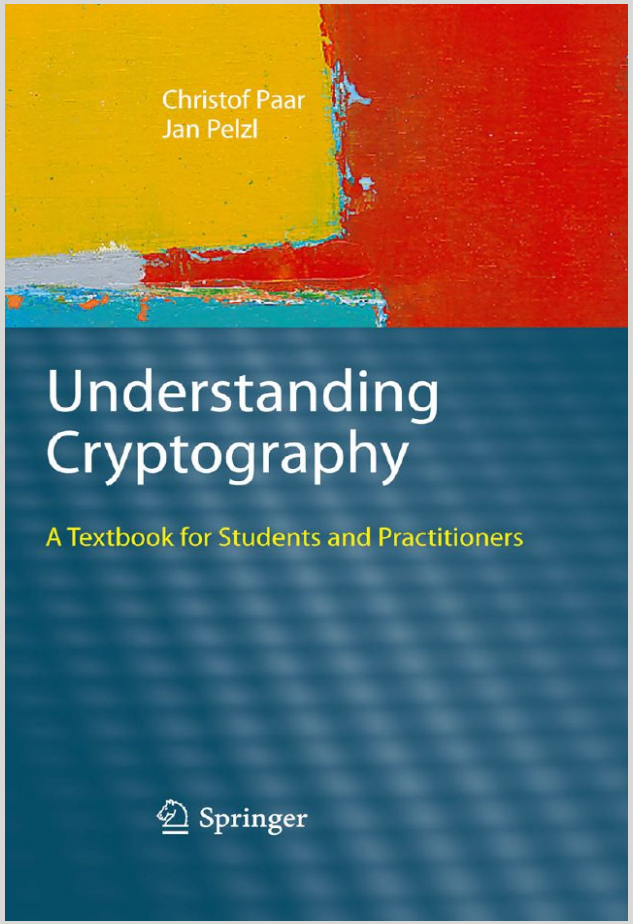
درس ششم



الگوریتم رمزنگاری DES

معرفی مـرجع

الگوریتم رمزنگاری DES



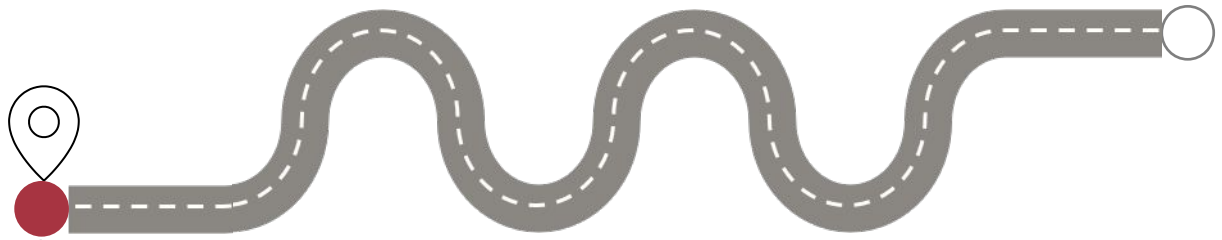
Paar, C., & Pelzl, J. (2009). Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media.

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 DES

- مقدمه
- معرفی DES
- چالش طول کلید
- راه‌کارهای افزایش طول کلید
- سایر ضعف‌های DES
- جمع‌بندی مطالب



مقدمه

DES

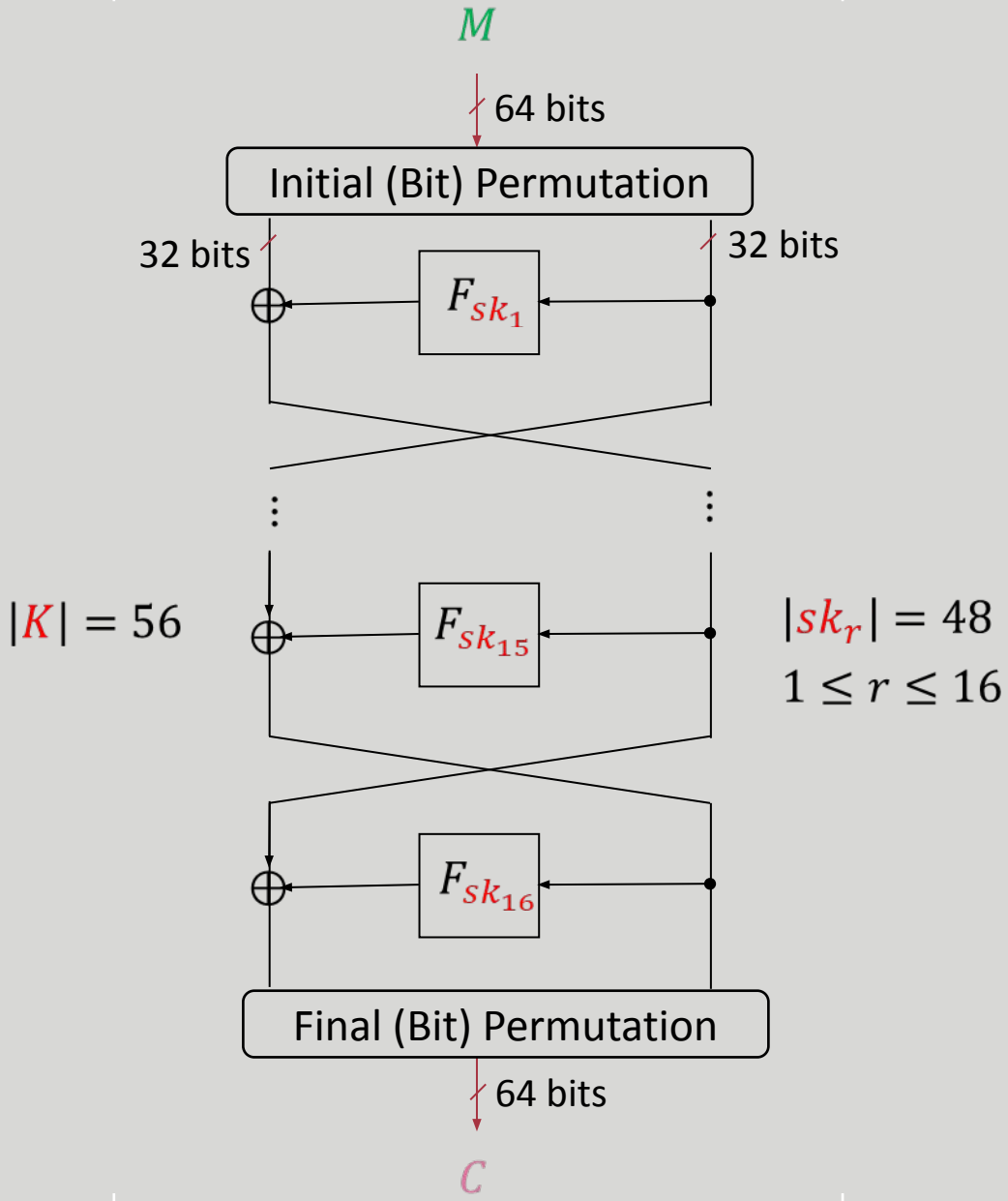
(Data Encryption Standard)

- DES در سال 1977 توسط محققین IBM طراحی شد.
- در سال 1979 توسط مؤسسه ملی فناوری و استانداردهای آمریکا، NIST (که در دهه ۷۰ میلادی NBS بود)، به عنوان استاندارد اعلام شد.
- در فرآیند نهایی‌سازی DES، سازمان امنیت ملی آمریکا (NSA) نقش موثری ایفا کرد.
- معیارهای طراحی DES توسط طراحان منتشر نشد.
- بررسی امنیت DES توسط رمزنگاران در دهه‌های ۸۰ و ۹۰ میلادی نقش به‌سزایی در پیشرفت علم در حوزه‌ی طراحی و تحلیل رمزهای قالبی داشت.
- هنوز در قالب الگوریتم رمزنگاری 3DES در برخی کاربردها استفاده می‌شود.

معرفی DES

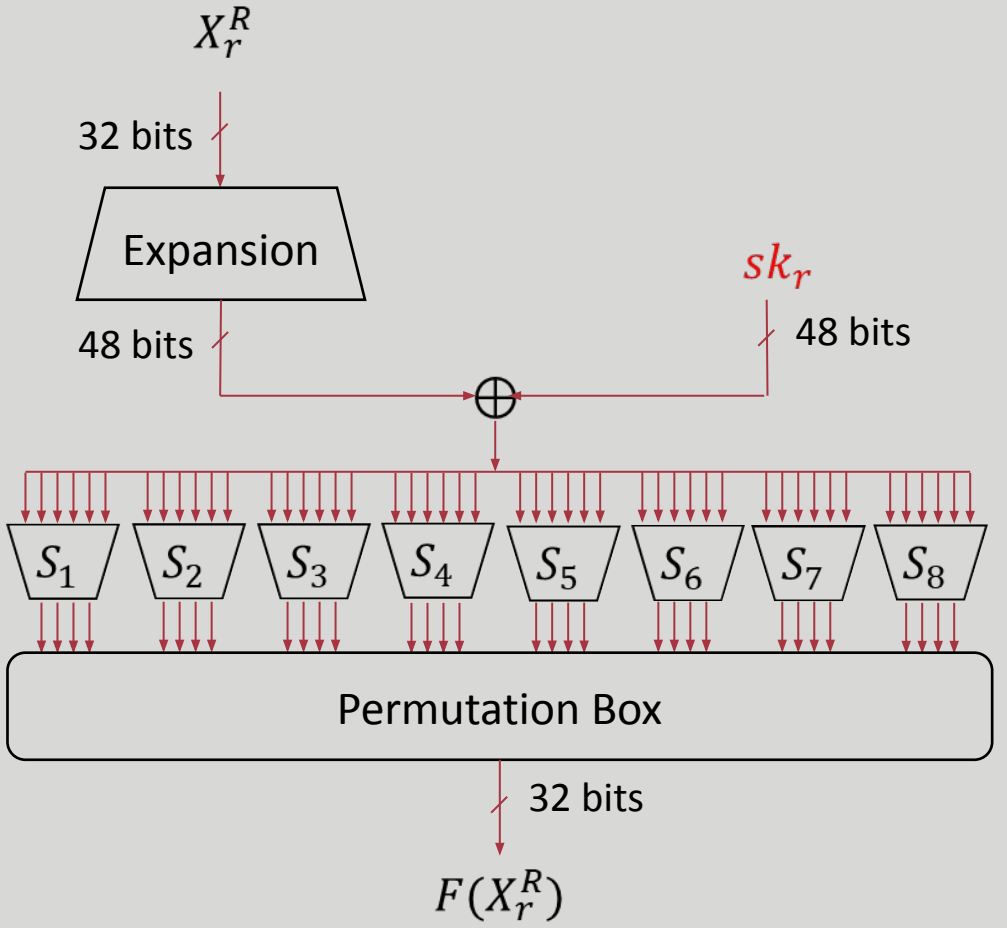
مقدمه

ساختار کلی DES



- DES یک رمز قالبی فیستلی ۱۶ دوری با طول قالب ۶۴ بیت است.
- در ابتدا و انتهای این رمز یک جایگشت بیتی وجود دارد که بودن یا نبودن آن تاثیری در امنیت ندارد!
- طول کلید 64 بیت است که شامل 56 بیت کلید مخفی و 8 بیت Parity Check است.
- ۱۶ زیرکلید ۴۸ بیتی با استفاده از طرح تولید زیرکلید از روی کلید مخفی تولید می‌شوند.

تابع دور DES



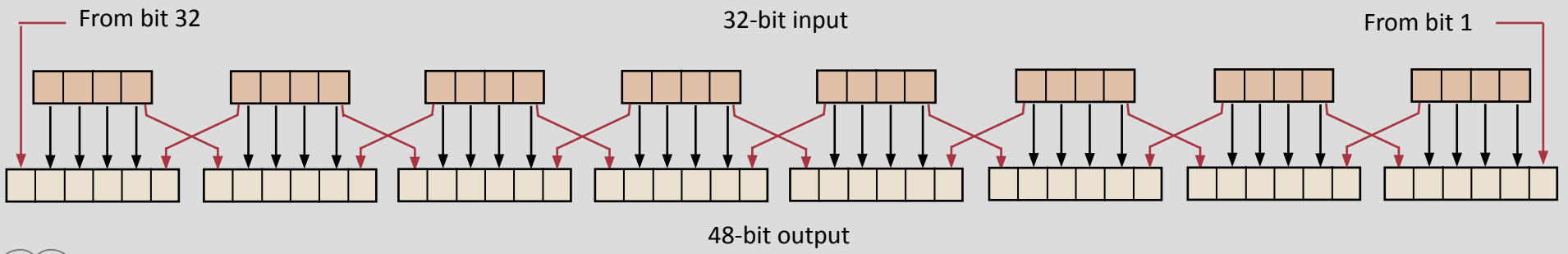
- تابع دور DES، به نام تابع Mangler نیز شناخته می‌شود.
- این تابع (F) طی چهار مرحله، ۳۲ بیت ورودی را به ۳۲ بیت خروجی منتقل می‌کند.

1. ابتدا ۳۲ بیت ورودی به ۴۸ بیت بسط داده می‌شوند.
2. سپس ۴۸ بیت زیرکلید به صورت Xor بیتی با حالت (State) جمع می‌شوند.
3. ۴۸ بیت حاصل شده، به ۸ دسته‌ی ۶ بیتی تقسیم شده و هر ۶ بیت از یک تابع غیرخطی به نام جعبه‌ی جانشانی عبور می‌کنند تا در نهایت به ۴ بیت تبدیل شوند.
4. در انتها بر روی این ۳۲ بیت یک جایگشت اعمال می‌شود.

تابع بسط ورودی

(Expansion)

- ۳۲ بیت ورودی به ۸ دسته‌ی ۴ بیتی تقسیم می‌شوند.
- سپس با کپی‌برداری از بیت‌های اول و آخر هر دسته‌ی چهارتایی، ۴۸ بیت خروجی تولید می‌شود.



Picture's Source: Behrouz Forouzan

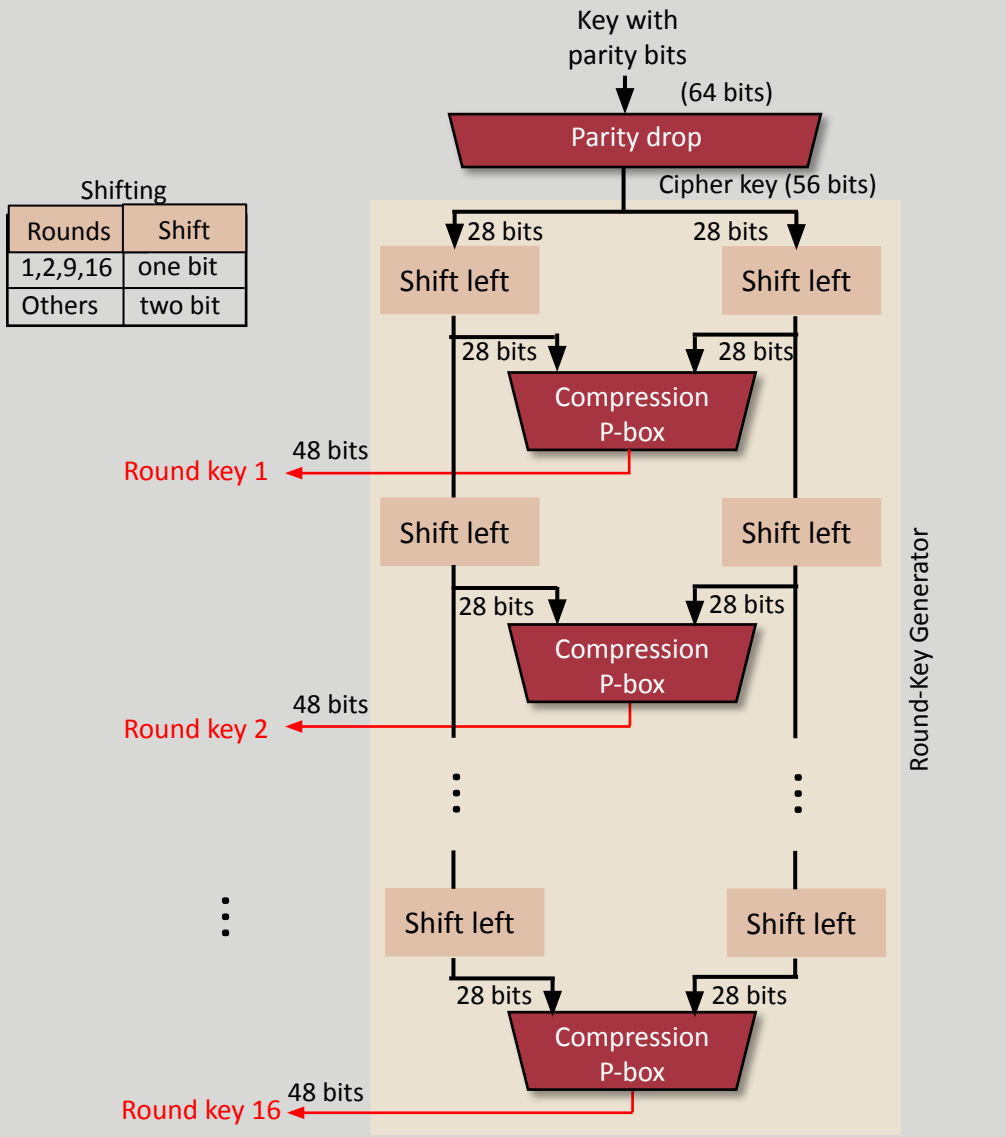
تولید کلیدهای دور

(Key schedule)

- 56 بیت اصلی کلید به دو بخش 28 بیتی تقسیم می‌شود.
- برای تولید زیرکلید 48 بیتی هر دور:

1. با توجه به شماره‌ی دور، یک یا دو بیت شیفت دورانی به بخش‌های ۲۸ بیتی اعمال می‌شود.
2. در نهایت این بیت‌ها از یک جایگشت عبور کرده و طی الگوریتمی مشخص، ۴۸ بیت از 56 بیت به عنوان زیرکلید انتخاب می‌شود.

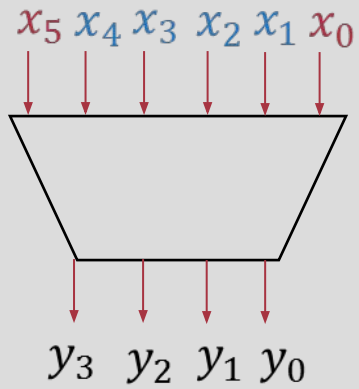
- ساختار تولید کلید به نحوی است که هر بیت کلید اصلی حدودا در ۱۴ دور استفاده می‌شود.



جعبه‌ی جانشانی

Substitution-box (Sbox)

- تابع دور شامل هشت جعبه‌ی جانشانی مختلف است.
- هر جعبه‌ی جانشانی 6 بیت ورودی را به 4 بیت خروجی تبدیل می‌کند.
- جعبه‌های جانشانی توسط طراحان DES به صورت جدول صحت‌هایی با ۴ سطر و ۱۶ ستون توصیف شده‌اند که دربردارنده‌ی مقادیر خروجی جعبه‌ی جانشانی هستند.
- بیت اول و آخر ورودی، شماره‌ی سطر و چهار بیت وسط ورودی شماره‌ی ستون را مشخص می‌کنند.

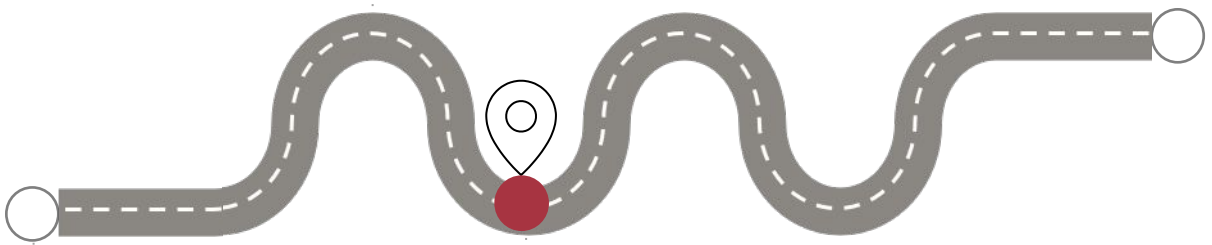


x_5x_0 \ $x_4x_3x_2x_1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	6	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	9	7	5	11	3	14	10	0	6	13

معیارهای طراحی و انتخاب اجزاء

- طراحی DES به گونه‌ای است که هر بیت خروجی دور پنجم به تمامی بیت‌های متن اصلی و کلید وابسته است.
- به خاطر وجود جعبه‌های جانشانی، توصیف جبری DES ساده نیست.
- نحوه‌ی انتخاب اجزاء به‌کاررفته (به خصوص جعبه‌های جانشانی)، تعداد دورها، و ... توسط طراحان اعلام نشد.
- ظاهرا منع آژانس امنیت ملی آمریکا دلیل عدم افشای نحوه‌ی طراحی DES بوده است.
- با تحقیقاتی که در سالیان بعد صورت گرفت، به برخی از سوالات پاسخ داده شد.
- به عنوان مثال، در ابتدای دهه‌ی 90 میلادی با معرفی تحلیل تفاضلی، چرایی و چگونگی نحوه‌ی انتخاب جعبه‌های جانشانی و همچنین تعداد دورها (تا حدودی) مشخص شد.

معرفی DES



مقدمه

چالش طول کلید

طول کلید کوتاه DES

- مهم‌ترین ضعف DES از دیدگاه رمزنگاران، طول کلید کوتاه آن بود.
- در سناریوی متن اصلی معلوم، اگر فرض کنیم که مهاجم به زوج (M,C) دسترسی دارد، می‌توان تمامی کلیدها را امتحان کرد.
- بنابراین الگوریتم (مستقل از ویژگی‌های اجزاء و ساختار آن) با پیچیدگی 2^{56} قابل شکستن است.
- ظاهرا در نسخه‌ی اولیه، IBM طول کلید را بیشتر در نظر گرفته بود، که با درخواست آژانس امنیت ملی آمریکا طول کلید کاهش می‌یابد.

The **United States Senate Select Committee on Intelligence** (1978):

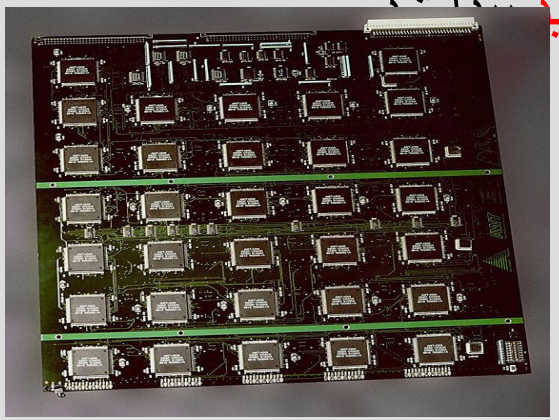
- "**NSA** convinced **IBM** that a reduced key size was sufficient; indirectly assisted in the development of the **S-box** structures; and certified that the final **DES** algorithm was, to the best of their knowledge, free from any statistical or mathematical weakness."
- "**NSA** did not tamper with the design of the algorithm in any way. **IBM** invented and designed the algorithm, made all pertinent decisions regarding it, and concurred that the agreed upon key size was **more than adequate** for all commercial applications for which the **DES** was intended."

روش Hellman برای جست‌وجوی کامل DES

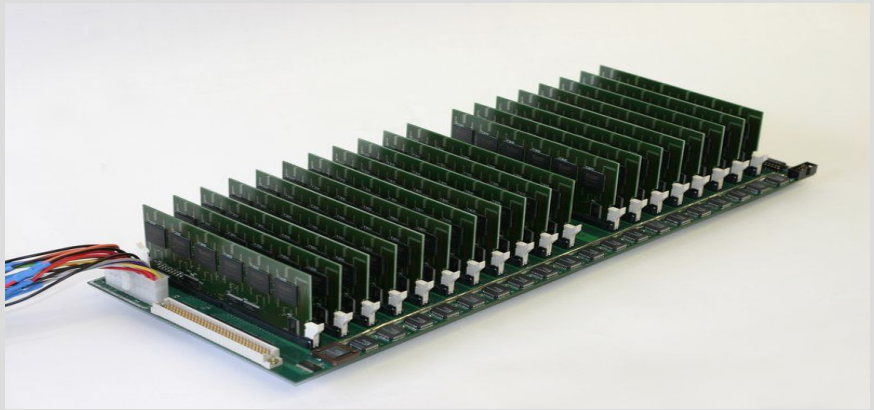
- Hellman در سال ۱۹۷۷ ادعا کرد که می‌توان با استفاده از یک میلیون تراشه در مدت زمان ۲۰ ساعت کل فضای کلید DES (2^{56} حالت) را امتحان کرد.
- وی هم‌چنین هزینه‌ی تقریبی این کار را در حدود بیست میلیون دلار عنوان کرد.
- در پاسخ، تیم طراح عملی بودن خرید یا فروش یک میلیون تراشه و هم‌چنین صرفه‌ی اقتصادی این کار را زیر سوال برد!
  - علی‌الخصوص این‌که کاربرد DES اغلب در موارد تجاری و غیرحساس است و این میزان هزینه‌کرد برای غلبه بر آن توجیه اقتصادی ندارد.
- هم‌چنین عنوان شد که در صورت استفاده از تراشه‌های کم‌تر نیز، به زمان بیش‌تری نیاز خواهد بود که می‌تواند منجر به خراب شدن تراشه‌ها قبل از پیدا کردن کلید شود (با توجه به تکنولوژی موجود در دهه‌ی ۸۰ میلادی).

جست‌وجوی کامل DES در دهه ۹۰

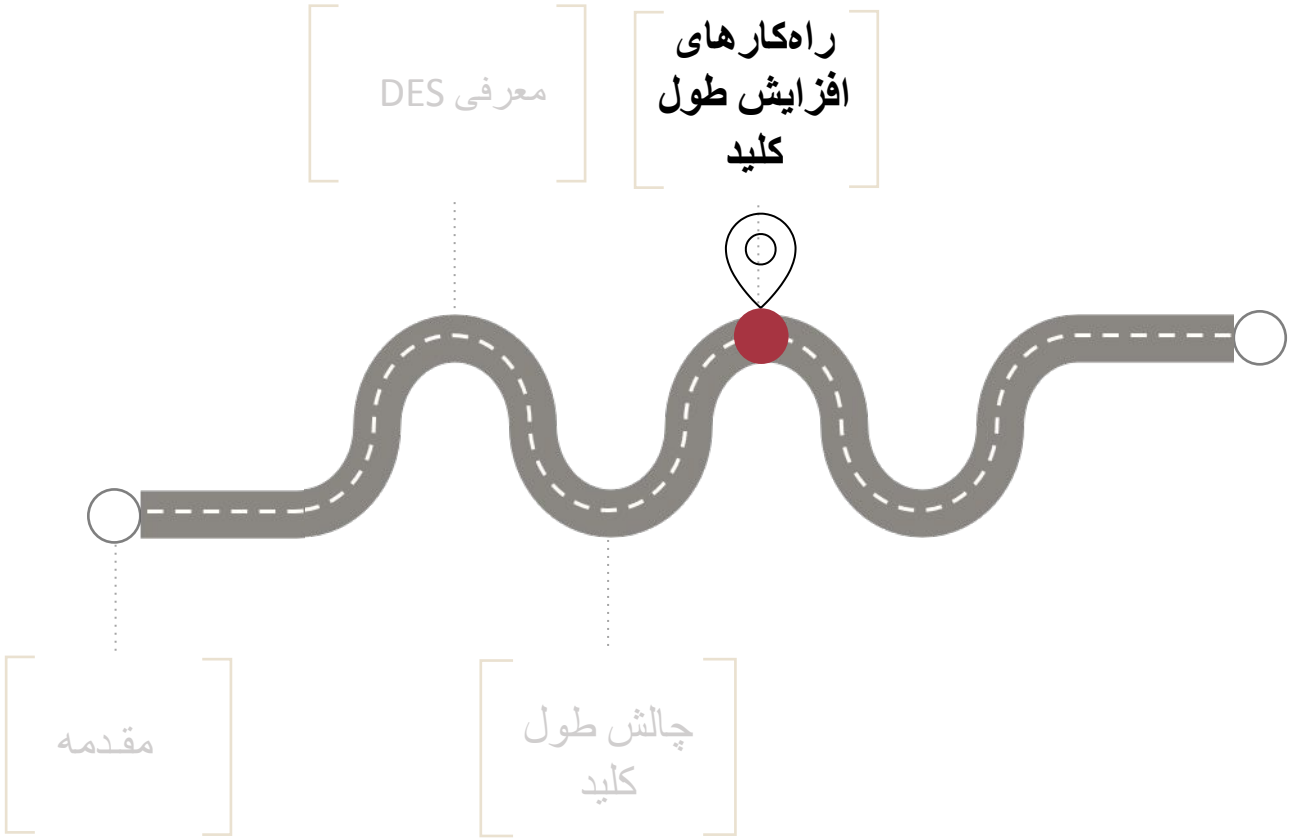
- ۱۹۹۳: Wiener با استفاده از فناوری 0.8 میکرو یک تراشه طراحی کرد که در صورت تولید انبوه به هزینه‌ای بالغ بر 1 میلیون دلار نیاز داشت.
- 1997: محققین پروژه‌ی DESCHALL توانستند کلید DES را در مسابقه‌ای که توسط شرکت RSA برگزار شده بود ظرف مدت 96 روز پیدا کنند.
- 1998: DES-Cracker با هزینه‌ی 250 هزار دلار توسط Electronic Frontier Foundation (EFF) ساخته شد.
- 1998: با استفاده از COPACOBANA طی مدت یک هفته و هزینه‌ی 10 هزار دلار کلید پیدا شد.



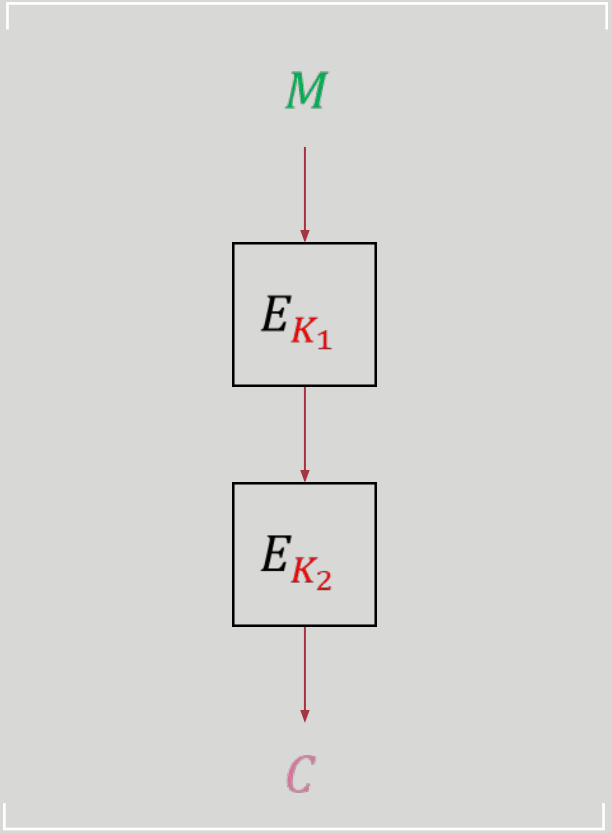
DES-cracker



COPACOBANA


معرفی DES
راه‌کارهای افزایش طول کلید
مقدمه
چالش طول کلید

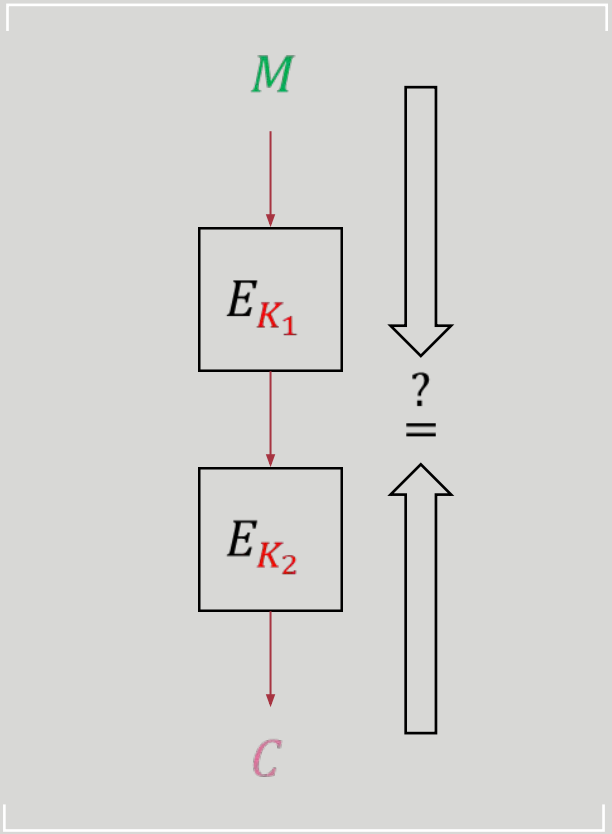
روش اول: Double DES



- یکی از راهکارهای ممکن برای جبران طول کلید کوتاه DES، استفاده از دو عملیات رمزنگاری DES تحت دو کلید است که اصطلاحا روش Double DES نامیده می‌شود.
- در این حالت پیچیدگی حمله‌ی جست‌وجوی جامع به $2^{56+56} = 2^{112}$ افزایش پیدا می‌کند.
- اما مقاومت در برابر حمله‌ی جست‌وجوی جامع لزوما به معنای امنیت در مقابل سایر حملات نیست!
- یکی از حملات مشهور به این ساختار با استفاده از تحلیل ملاقات در میانه است.

تحلیل ملاقات در میانه

(Meet-in-the-Middle Attack)

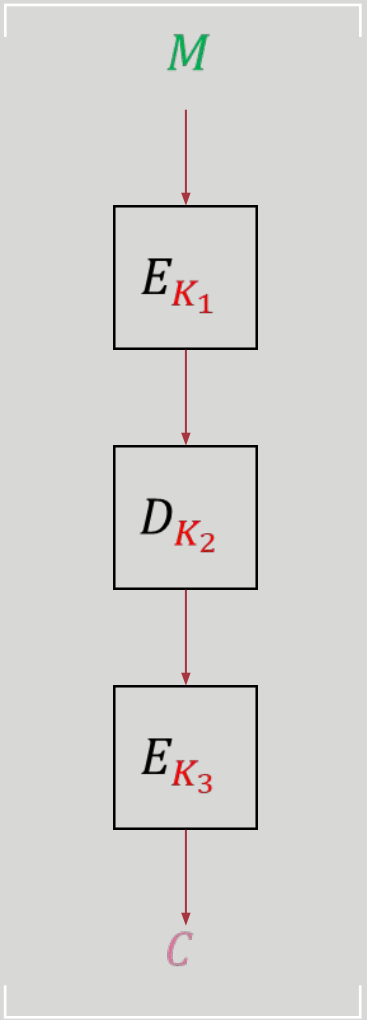


- فرض کنید که زوج متن معلوم (M,C) داده شده است.
- **گام اول:** متن اصلی M را به ازای تمامی کلیدهای ممکن K_1 رمز کرده و در یک جدول (H) ذخیره می‌کنیم.
- سپس جدول را براساس مقدار (M) E_{K_1} مرتب می‌کنیم.
- **گام دوم:** کلید K_2 را حدس زده و متن رمز شده C را رمزگشایی کرده و با مقادیر پیش‌محاسبه شده در جدول H مقایسه می‌کنیم.
- اگر مقدار محاسبه شده برابر با مقدار موجود در سطر K_1 جدول باشد، احتمالا زوج (K_1,K_2) کلید صحیح است. در غیر این صورت قطعاً غلط است.

پیچیدگی تحلیل ملاقات در میانه

- پیچیدگی زمانی هر کدام از گام‌های اول و دوم برابر با 2^{56} است.
- از آنجایی‌که دو مرحله از هم مستقل هستند، پیچیدگی زمانی کل برابر با $2^{56} + 2^{56} = 2^{57}$ است.
- برای ذخیره‌ی مقادیر پیش‌محاسبه‌شده در جدول H به $2^{56} \times 8 = 2^{59}$ بایت نیاز است (پیچیدگی حافظه).
- احتمال برابری دو مقدار (تصادفی) ۶۴ بیتی با یکدیگر برابر 2^{-64} است.
- بنابراین انتظار داریم حدود $2^{112-64} = 2^{48}$ کاندید برای کلید باقی بماند.
- کاندیدهای باقی‌مانده می‌توانند با استفاده از یک زوج متن معلوم دیگر چک شوند تا کلید مخفی به دست بیاید.
- بنابراین پیچیدگی داده برابر با دو متن معلوم است.
- بنابراین روش Double DES امنیت را افزایش نمی‌دهد.

روش دوم: 3DES(Triple DES)



- ساختار پیشنهادی NIST: استفاده از سه عملیات متوالی DES به صورت زیر است:
  1. رمزگذاری با کلید K_1
  2. رمزگشایی با کلید K_2
  3. رمزگذاری با کلید K_3
- در حالتی‌که $K_1 = K_2 = K_3 = K$ باشد، 3DES معادل عملیات رمزگذاری DES تحت کلید K است چراکه در این حالت، ورودی رمزگذاری دوم پیام اصلی است:

$$D_K\left(E_K(M)\right) = M$$

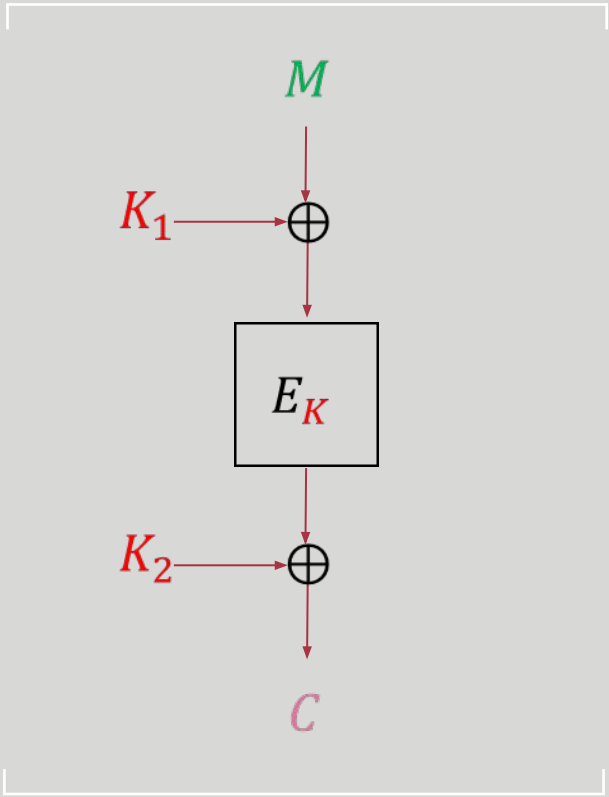
- در نتیجه برای افرادی که قبلا از DES استفاده می‌کرده‌اند ساختار مناسبی است چراکه راحت‌تر می‌توانند استاندارد جدید را پذیرفته و خود را با آن تطبیق دهند.

روش دوم: 3DES(Triple DES)

... ادامه

- تحلیل ملاقات در میانه از لحاظ تئوری به 3DES نیز قابل اعمال است، اما پیچیدگی زمانی آن حدود 2^{100} است که در عمل قابل اجرا نیست.
- ضعف عمده‌ی 3DES این است که (به خصوص در پیاده‌سازی‌های نرم‌افزاری)، کارایی را کاهش می‌دهد.
- 3DES در بسیاری از کاربردهای عملی مورد استفاده قرار گرفت.
- با معرفی استاندارد جدید AES، 3DES به مرور جایگزین شده است هرچند که هنوز در برخی از کاربردهای عملی استفاده می‌شود.
- در سال ۲۰۱۶ نشان داده شد که طول قالب کوتاه 3DES می‌تواند در TLS مورد استفاده قرار گیرد (حمله‌ی Sweet32).
- در سال ۲۰۱۷، NIST منسوخ شدن 3DES و لزوم جایگزینی آن با رمزهای قالبی امن جدید را اعلام کرد.

روش سوم: DESX

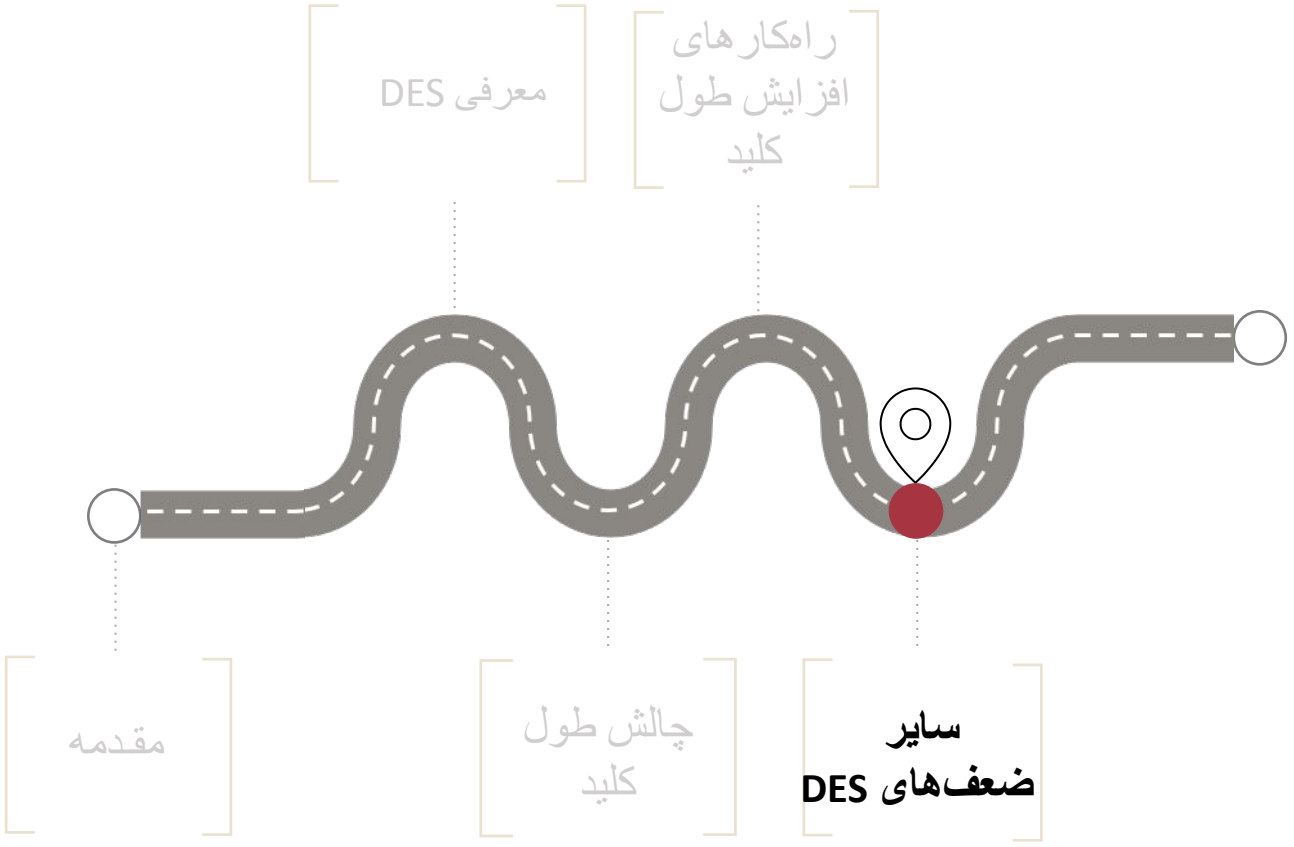


- در سال ۱۹۸۴، ساختار دیگری نیز برای برطرف کردن طول کلید کوتاه DES توسط Rivest پیشنهاد شد که به DES-X یا DESX مشهور است.
- در این ساختار قبل و بعد از عملیات DES، کلیدهای دیگری با حالت، Xor می‌شوند:

$$\text{DESX} = K_2 \oplus \text{DES}_K(M \oplus K_1)$$

که $|K_1| = |K_2| = 64$.

- طول کلید در این حالت از 56 بیت به $56 + (64 \times 2) = 184$ بیت افزایش پیدا می‌کند.
- با فرض امن بودن DES می‌توان اثبات کرد که این طرح در صورت دسترسی مهاجم به 2^m متن معلوم، با پیچیدگی زمانی سریع‌تر از 2^{119-m} شکسته نمی‌شود.


معرفی DES
راه‌کارهای افزایش طول کلید
مقدمه
چالش طول کلید
سایر ضعف‌های DES

ویژگی مکمل بودن

(Complementation Property)

- الگوریتم رمزنگاری DES دارای ویژگی‌ای است که عموما به عنوان ویژگی مکمل بودن شناخته می‌شود:

$$\mathrm{DES}_K(M) = C \Rightarrow \mathrm{DES}_{\bar{K}}(\bar{M}) = \bar{C}$$

- با وجود این ویژگی غیرتصادفی، در مدل متن اصلی منتخب، امنیت از 56 بیت به 55 بیت کاهش پیدا می‌کند.
- در تمرین‌های درس از شما خواسته می‌شود که ویژگی مکمل بودن را برای DES اثبات و با استفاده از آن حمله‌ای با پیچیدگی 2^{55} ارائه کنید ☺

وجود کلیدهای ضعیف در DES

- در ساختار فیستلی رمزگشایی با استفاده از همان الگوریتم رمزگذاری و صرفا با تغییر ترتیب زیرکلیدها انجام می‌شود.
- اگر تمامی زیرکلیدها باهم برابر باشند، تابع رمزگذاری و رمزگشایی یکسان می‌شوند!
- به کلیدی که زیرکلیدهای حاصل از آن برابر شوند، کلید ضعیف گفته می‌شود.
- DES چهار کلید ضعیف دارد!

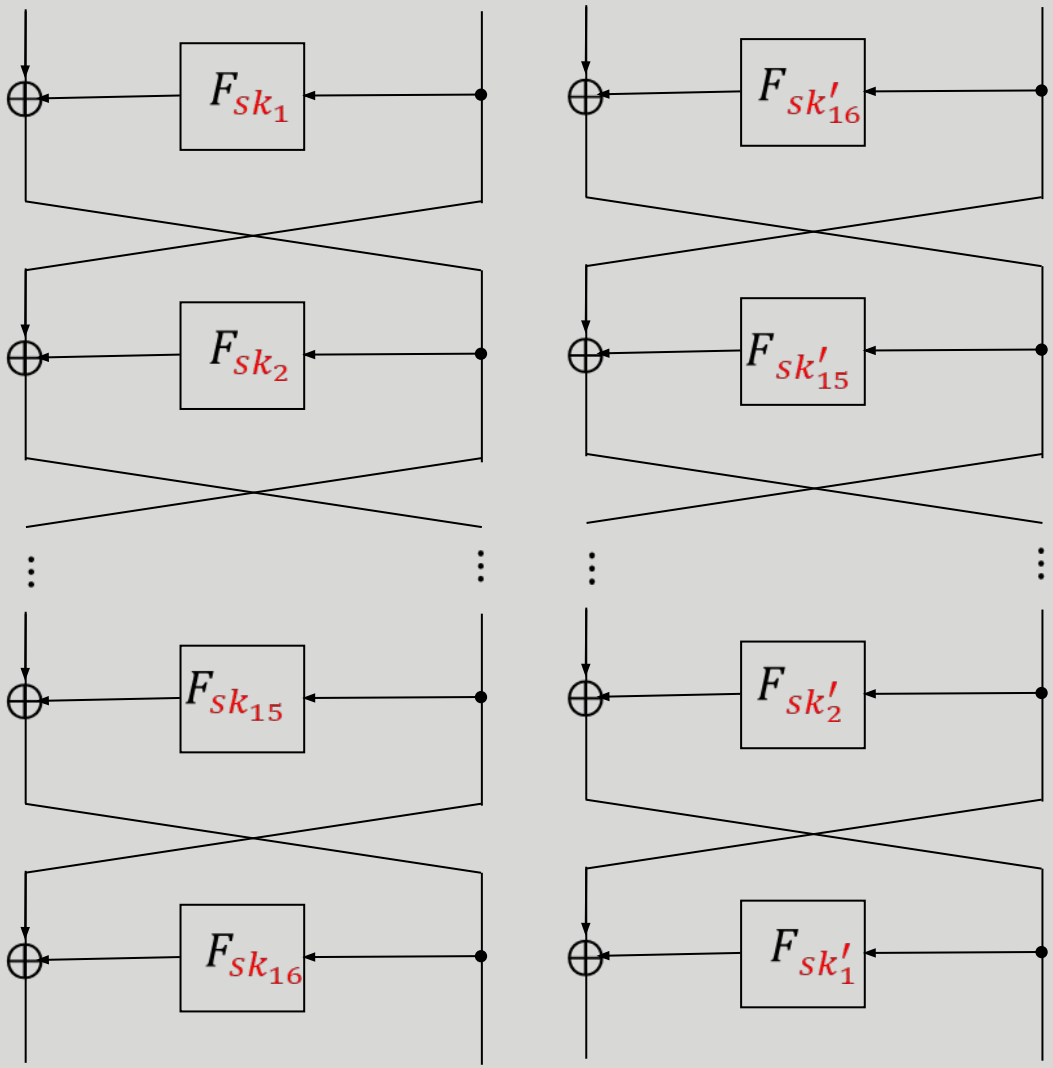
کلید مخفی (۵۶ بیتی)	کلید ۶۴ بیتی با احتساب Parity Check
0x000000000 0000000	0x0101010101010101
0xFFFFFFFFF FFFFFFF	0xFEFEFEFEFEFEFEFE
0xFFFFFFFFF 0000000	0xE0E0E0E0F1F1F1F1
0x0000000 FFFFFFF	0x1F1F1F1F0E0E0E0E

کلیدهای شبه‌ضعیف

- فرض کنید زیرکلیدهای حاصل از کلید K را با $sk_1, sk_2, \ldots, sk_{16}$ و زیرکلیدهای حاصل از کلید K' را با $sk'_1, sk'_2, \ldots, sk'_{16}$ نمایش دهیم.
- فرض کنید که رابطه‌ی زیر بین زیرکلیدهای K و K' برقرار باشد:

$$sk_i = sk'_{16-i} \text{ for } 1 \le i \le 16$$

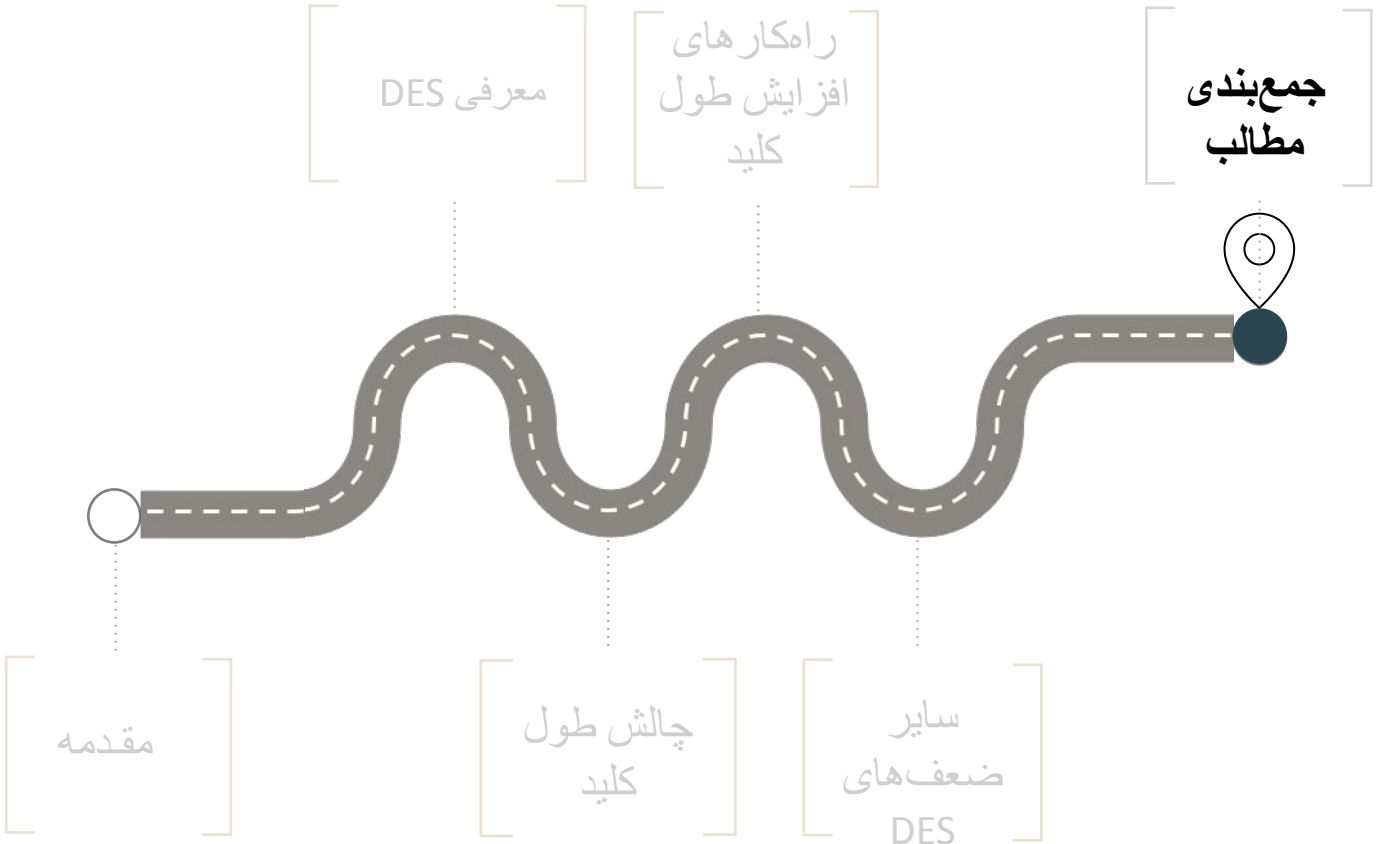
- در این صورت چه اتفاقی می‌افتد؟
- رمزگذاری (رمزگشایی) تحت کلید K، برابر با رمزگشایی (رمزگذاری) تحت کلید K' می‌شود.
- به زوج کلیدهای K و K' که چنین رابطه‌ای دارند، زوج کلید شبه‌ضعیف گفته می‌شود.



کلیدهای شبه‌ضعیف در DES

- الگوریتم DES شش جفت کلید شبه‌ضعیف دارد.
- در عمل، احتمال آن‌که وجود چنین کلیدهایی منجر به تهدید جدی شوند بسیار کم است.
- اما به لحاظ نظری نشان‌دهنده‌ی یک ضعف در ساختار DES محسوب می‌شوند.

First key	Second key
0x011F011F010E010E	0x1F011F010E010E01
0x01E001E001F101F1	0xE001E001F101F101
0x01FE01FE01FE01FE	0xFE01FE01FE01FE01
0x1FE01FE00EF10EF1	0xE01FE01FF10EF10E
0x1FFE1FFE0EFE0EFE	0xFE1FFE1FFE0EFE0E
0xE0FEE0FEF1FEF1FE	0xFEE0FEE0FEF1FEF1


جمع‌بندی مطالب
راه‌کارهای افزایش طول کلید
معرفی DES
سایر ضعف‌های DES
چالش طول کلید
مقدمه

جمع‌بندی مطالب

- الگوریتم رمزنگاری DES اولین رمز قالبی استاندارد و پرکاربرد است.
- مهمترین نقاط ضعف DES طول کلید کوتاه و روشن نبودن معیارهای طراحی آن می‌باشد.
- به خاطر ضعف‌های DES، الگوریتم رمزنگاری جدید AES استاندارد شده و در بیشتر کاربردهای امروزی مورد استفاده قرار می‌گیرد.
- و البته الگوریتم 3DES نیز هنوز در برخی از کاربردها استفاده می‌شود.
- در درس بعدی با الگوریتم AES آشنا خواهیم شد.