

## تکلیف دوم

مهدی حقوردی

۳۰ فروردین ۱۴۰۳

۱

با توجه به سیستم رمزنگاری DES به سوالات زیر پاسخ دهید.

(آ) تعداد کل عملیات های xor را بدست آورید.

از آنجایی که DES یک ساختار فیستلی ۱۶ دوری است، در بیرون از تابع F، ۱۶ تا xor قرار دارد. و چون درون تابع F پس از عملیات extend یک بار با کلید xor انجام می گیرد پس اینجا هم ۱۶ تا عملیات xor داریم و در مجموع ۳۲ عملیات xor.

(ب) هدف از s-box ها را بنویسید.

نوشتن رابطه ی جبری برای بیت های خروجی بر حسب بیت های ورودی و کلید به دلیل وجود s-box بسیار دشوار است.

(ج) پیچیدگی حمله ی جست و جوی جامع به این سیستم از چه مرتبه ای می باشد؟

کلید DES، ۶۴ بیتی است که ۸ بیت آن بیت های parity هستند پس کلید مخفی آن تنها ۵۶ بیت طول دارد ← جستجوی کامل در DES از مرتبه ی  $2^{56}$  است.

(د) دلیل استفاده از expansion s-box در DES Function چیست؟

کلید ۵۶ بیتی DES توسط Key Scheduler به ۱۶ کلید ۴۸ بیتی تبدیل می شود و از آنجایی که طول بلاک DES ۶۴ بیت است و در ساختار فیستل تنها ۳۲ بیت آن به داخل تابع F می رود باید ۳۲ بیت ورودی را به ۴۸ بیت گسترش بدهیم تا بتوانیم آن را با کلید xor کنیم.

(ه) اگر خروجی سیستم رمزنگاری به یک سیستم رمزنگاری دیگر داده شود، چه تغییری در امنیت آن حاصل می شود؟ (double des) اگر این کار سه بار تکرار شود چگونه؟ (triple des)

• double des

در این حالت برای شکستن می توان از حمله ی تطابق در میانه استفاده کرد که مرتبه ی آن از  $2^{112}$  به  $2^{57}$  تقلیل می یابد.

• triple des

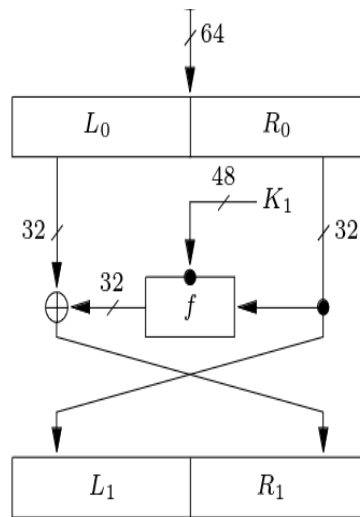
در این حالت هم (با استفاده از حمله ی تطابق در میانه) مرتبه بجای  $2^{168}$  می شود:  $2^{112}$  که البته در عمل قابل انجام نیست. در سال ۲۰۱۷ NIST منسوخ شدن 3DES را اعلام کرد.

و) ویژگی مکمل بودن این سیستم را ثابت کنید و توضیح دهید در آن صورت حمله به این سیستم از چه مرتبه‌ایست و چرا؟

خاصیت مکمل بودن DES:

$$\text{DES}_K(M) = C \Rightarrow \text{DES}_{\bar{K}}(\bar{M}) = \bar{C} \quad (1)$$

(برای اثبات فقط یک دور را در نظر میگیریم) با توجه به ساختار فیستلی DES ورودی ابتدا به دو قسمت تقسیم میکند و سپس نیمه‌ی راست را (درون تابع F) با کلید xor می‌کند و سپس خروجی را با قسمت سمت چپ xor می‌کند.



که یعنی:

$$\begin{cases} P = L_0.R_0 \\ L_1 = R_0 \\ B = f(R_0 \oplus K_1) \\ R_1 = L_0 \oplus B \end{cases} \Rightarrow C = L_1.R_1 \quad (2)$$

حال اگر  $P$  و  $K$  را not کنیم:

$$\begin{cases} \bar{P} = \bar{L_0.R_0} \\ L_1 = \bar{R_0} \\ B = f(\bar{R_0} \oplus \bar{K_1}) \\ R_1 = \bar{L_0} \oplus B \end{cases} \Rightarrow \bar{C} = \bar{L_1.R_1} \quad (3)$$

پس در نتیجه:

$$E_K(P) = C \iff E_{\bar{K}}(\bar{P}) = \bar{C} \quad (4)$$

با استفاده از یک کلید رمز واحد، هر یک از تبدیلات زیر را بر متن آشکار که تنها در بیت اول با هم تفاوت دارند، اعمال کنید. تعداد بیت‌های تغییر یافته پس از هر تبدیل را پیدا کنید. هر تبدیل را بطور مستقل اعمال کنید. در مورد اثر بهمنی پس از هر تبدیل بطور مستقل و سپس اثر بهمنی پس از اعمال یک راند توضیح دهید.

برای نوشتن این سوال هر عملیات AES را در پایتون پیاده سازی کردم که source code آن در پوشه‌ی AES همراه تکلیف ارسال شده است. پاسخ هر بخش در تصویری که جلوی شما نوشته شده است نوشته شده است.

## ۱.۲ توضیح تصاویر

ON: Op Name

P1: 0000000000

C1: 1000000001

P2: 1000000000

C2: 0111001110

C1: 1000000001

C2: 0111001110

ON: Op Name

Changed bit No.	Bit change ratio
Cnt	8
ratio	80%

اول از همه نام عملیات در بالای تصویر نوشته شده است،

سپس متن آشکار و متن تغییر یافته و کاراکترهایی که تغییر یافته‌اند نشان داده شده‌اند،

دوباره همین کار روی متن آشکاری که بیت اول آن فرق کرده است تکرار شده است،

سپس تفاوت‌های بین دو متن تغییر یافته نوشته شده،

و در آخر در جدولی تعداد بیت‌های تغییر یافته و درصد تغییر یافتن متن رمز شده‌ی دوم نوشته شده است.

## ۲.۲ پاسخ‌ها

(آ) SB: Sub Byte تصویر ۱(آ)

(ب) SR: Shift Row تصویر ۱(ب)

(ج) MC: Mix Columns تصویر ۱(ج)

(د) ARK: Add Round Key تصویر ۱(د)

(ه) FR: Full Round تصویر ۱(ه)

همانطور که مشاهده شد، اثر بهمنی در عملیات‌های مختلف درصد کمی داشته و در یک دور (آن هم در این مورد خاص که اولین بیت تغییر کرده است) به ۱۸% رسید. اگر ما با کلیدی ۱۲۸ بیتی و عملیات کامل رمزنگار AES که شامل ۱۶ دور است، (طبق مستندات) اثر بهمنی به نزدیک حداکثر آن، یعنی ۵۰% می‌رسد.

از بین مدهای عملیاتی ECB، CBC، OFB، CFB و CTR در کدام یک امکان افزایش سرعت در عمل رمزگذاری با استفاده از parallel processing یا پردازش موازی وجود دارد؟  
مدهای: CTR • ECB

SR: Shift Row

P1: 0000110011001010010101100111000110000100010010101111010000110111000100100101101101

C1: 0000110011000010011111001101001100000011101010010010100101101101001100100001001010010011

P2: 1000110011001010010100101100111000110000100010010101011110100100110001000100100101101101

C2: 1000110011000010011111001101010011100000011101010010010100101101101001101100001001010010011

P3: 00001100110000100111110011010100111000000111010100100101001011101000101100100001001010010011

C3: 100011001100001001111100110101001110000001110101001001010010111010011011010010001001010010011

SR: Shift Row

Changed bit No.	Bit change ratio
Cnt	1
ratio	0.78125%

SB: Sub Byte

P1: 0000110011001010010101100111000110000100010010110101111101001001110001000110111000100100101101101

C1: 111111001101001011110101001100000011001001100000011001001010100111001111011001100010001000111100

P2: 1000110011001010010100100110001100001000010010110101010111010000011101011111010010011000100100101101101

C2: 011001000111010010111101011001100000011100100110000001110010010101001110011111011100110000110001111100

P3: 11111100111010011110101100110000001110010011000000111001001010100111001111101110011000011000111100

C3: 011001000111010010111101011001100000011100100110000001110010010101001110011110111001100010001000111100

SB: Sub Bytes

Changed bit No.	Bit change ratio
Cnt	4
ratio	3.125%

SR: Shift Row (ب)

SB: Sub Byte (ڀ)

ARK: Add Round Key (⌢)

MC: Mix Column ( $\tau$ )

FR: Full Round (•)

Y