

DES:

الگوریتم DES در دهه ۷۰ میلادی در آمریکا به عنوان یک استاندارد کدگذاری مطرح شد. این الگوریتم این گونه عمل می کند که رشته ای از متن اصلی با طول ثابت را به عنوان ورودی می گیرد و پس از انجام یک سری اعمال پیچیده روی آن خروجی را که طولی برابر طول ورودی دارد تولید می کند DES. هم چنین از یک کلید برای ایجاد رمز استفاده می کند و تنها کسانی قادر به رمزگشایی خواهند بود که مقدار کلید را می دانند. اگرچه تحلیل هایی که درباره DES انجام شده است از هر روش رمز قطعه ای دیگری بیشتر است ولی عملی ترین حمله علیه این الگوریتم جستجوی جامع فضای کلید است. سه حمله تئوریک برای این الگوریتم وجود دارند که زمان کمتری نسبت به جستجوی جامع فضای کلید نیاز دارند ولی این روشها در عمل امکان پذیر نیستند.

با شکسته شدن الگوریتم DES این استاندارد در سال ۱۹۹۸ تمدید نشد و در سال ۲۰۰۱، الگوریتم AES به عنوان استاندارد جایگزین آن تصویب شد. این الگوریتم مانند DES یک الگوریتم رمز قطعه ای است ولی بر خلاف DES از ساختار فیستل استفاده نمی کند. تا سال ۲۰۰۶ تنها حمله مؤثر علیه الگوریتم AES حمله side channel بوده است. در ژوئن سال ۲۰۰۳ دولت آمریکا اعلام کرد که از AES می توان برای حفاظت از اطلاعات رده بندی شده و سری نیز استفاده کرد. برای اطلاعات فوق سری و محرمانه باید از کلیدهایی با طول ۱۹۲ یا ۲۵۶ بیت استفاده کرد.

در سال ۱۹۷۲ مؤسسه بین المللی استاندارد و فناوری آمریکا اعلام کرد که به یک الگوریتم برای حفاظت از اطلاعات غیر رده بندی شده خود نیاز دارد. این الگوریتم می بایست ارزان، قابل دسترس و بسیار مطمئن می بود. در سال ۱۹۷۳، NIST فراخوانی برای چنین الگوریتمی اعلام نمود ولی هیچ یک از الگوریتم هایی که در پاسخ به این فراخوان ارائه شدند شرایط لازم را نداشتند. دومین فراخوان در سال ۱۹۷۴ مطرح شد در این زمان IBM الگوریتم خود را مطرح نمود که به نظر می رسید می تواند نیازهای NIST را بر طرف کند. این الگوریتم به عنوان یک استاندارد فدرال در سال ۱۹۷۶ تصویب شد و در سال ۱۹۷۷ منتشر شد. با امکان پذیر شدن حمله جستجوی جامع فضای کلید برای این الگوریتم، سازمان ملی استاندارد و فناوری آمریکا در آغاز سال ۱۹۹۷ اعلام کرد که برای تدوین استاندارد پیشرفته رمزنگاری تلاشی را آغاز کرده است. در سپتامبر همان سال این سازمان به طور رسمی فراخوانی را برای ارائه الگوریتم های رمزنگاری اعلام نمود.

در کنفرانس اول، AES-1، ۱۵ الگوریتم کاندیدا انتخاب شدند، NIST از تمام دانشمندان و مؤسسه های علمی خواست که نظرات خود را در مورد این الگوریتم ها ارائه دهند. هم چنین NIST با کمک جامعه بین المللی رمزنگاری و تشکیل کمیته هایی، اقدام به بررسی قابلیت ها و توانایی های الگوریتم های ارائه شده نمود. در آگوست سال بعد، در سمینار دوم،

AES-2، پنج الگوریتم انتخاب و برای رقابت نهایی معرفی شدند. این الگوریتم‌ها عبارت بودند از - RC6 - Rijndael :
MARS - Twofish - Serpent

آخرین نظرات و انتقادات تا تاریخ ۱۵ مه ۱۹۹۹ جمع‌آوری شد و بالاخره در سمینار AES-3، پس از بررسی گزارش کمیته‌های بررسی کننده، الگوریتم Rijndael به عنوان الگوریتم استاندارد پذیرفته شد.

الگوریتم DES

در DES طول قطعات ۶۴ بیت است. کلید نیز شامل ۶۴ بیت است ولی در عمل تنها از ۵۶ بیت آن استفاده می‌شود و از ۸ بیت دیگر فقط برای چک کردن parity استفاده می‌شود. الگوریتم شامل ۱۶ مرحله مشابه است که هر مرحله یک دور ۴ نامیده می‌شود. متنی که قرار است رمزگذاری شود ابتدا در معرض یک جایگشت اولیه (IP) قرار می‌گیرد. سپس یک سری اعمال پیچیده وابسته به کلید روی آن انجام می‌شود و در نهایت در معرض یک جایگشت نهایی (FP) قرار می‌گیرد IP, FP. معکوس هم هستند FP عملی که توسط IP انجام شده است را خنثی می‌کند؛ بنابراین از جنبه رمزنگاری اهمیت چندانی ندارند و برای تسهیل نمودن بار کردن قطعات داده در سخت‌افزارهای دهه ۱۹۷۰ استفاده شدند ولی اجرای DES در نرم‌افزار را کند کردند. قبل از دور اصلی، داده به دو بخش ۳۲ بیتی تقسیم می‌شود که این دو نیمه به طور متناوب مورد پردازش قرار می‌گیرند این تقاطع به عنوان شکل فیستل شناخته می‌شود. ساختار فیستل تضمین می‌کند که رمزگذاری و رمزگشایی دو رویه کاملاً مشابه هم هستند و تنها تفاوت آنها این است که زیر کلیدها در زمان رمزگشایی در جهت معکوس رمزگذاری به کار برده می‌شوند؛ و بقیه الگوریتم در هر دو یکسان است که این امر پیاده‌سازی رابه خصوص در سخت‌افزار بسیار آسان می‌کند و دیگر نیازی به الگوریتم‌های متفاوت برای رمزگذاری و رمزگشایی نیست. تابعی که خروجی IP را می‌گیرد و پس از شانزده مرحله ورودی FP را فراهم می‌کند تابع F نامیده می‌شود. این تابع یک ورودی ۳۲ بیتی و یک ورودی ۴۸ بیتی دارد و یک خروجی ۳۲ بیتی تولید می‌کند. بلاک ورودی شامل ۳۲ بیت که نیمه سمت چپ را تشکیل می‌دهد و با L نشان داده می‌شود و به دنبال آن ۳۲ بیت دیگر که نیمه راست را تشکیل می‌دهد و با R نمایش داده می‌شود است. پس کل بلاک را می‌توان به صورت LR نمایش داد.

اگر K یک بلاک ۴۸ بیتی باشد که از کلید اصلی ۶۴ بیتی مشتق شده است و خروجی یک دور با ورودی LR و خروجی $L1R1$ به صورت زیر تعریف می‌شود $L1=R$ $R1=L \text{ XOR } F(R,K)$. اگر KS تابعی باشد که کلید ۶۴ بیتی KEY و یک عدد صحیح در محدوده ۱ تا ۱۶ را به عنوان ورودی می‌گیرد و کلید ۴۸ بیتی K_n را به عنوان خروجی تولید می‌کند به طوری که بیت‌های K_n از تغییر محل بیت‌های KEY حاصل شده‌اند داریم $K_n = KS(n, KEY)$:

KS را تابع key schedule می‌نامند؛ بنابراین در حالت کلی داریم $R_n = R_{n-1} \text{ XOR } f(R_{n-1}, K_n)$ برای رمزگشایی نیز داریم $R = L_1 \text{ XOR } f(L_1, K)$:

در نتیجه رمزگشایی با همان الگوریتمی که برای رمزگذاری استفاده شد انجام می‌شود و در هر مرحله همان K بیتی که به عنوان کلید برای رمزگذاری استفاده شده بود مورد استفاده قرار می‌گیرد بنابراین می‌توان نوشت $R_{n-1} = L_n$ $L_{n-1} = R_n$: $\text{XOR } f(L_n, K_n)$

برای محاسبات رمزگشایی $R_{16}L_{16}$ ورودی IP و $ROLO$ ورودی FP است. کلید شانزدهم در مرحله اول، کلید پانزدهم در مرحله دوم و به همین ترتیب کلید اول در مرحله شانزدهم مورد استفاده قرار می‌گیرد.

تابع F

بسط: در این مرحله با استفاده از یک جایگشت انبساطی ۳۲ بیت به ۴۸ بیت گسترش داده می‌شود.

ترکیب کلید: در این مرحله حاصل مرحله قبل با یک زیر کلید XOR می‌شود. شش کلید ۴۸ بیتی با استفاده از الگوریتم key schedule از کلید اصلی تولید می‌شود.

جایگزینی: بعد از ترکیب کلید هر قطعه داده به هشت بخش ۶ بیتی تقسیم می‌شود (قبل از پردازش توسط جعبه‌های جایگزینی (هر کدام از $S\text{-box}$ ها ورودی ۶ بیتی خود را با استفاده از یک تبدیل غیر خطی که به شکل یک جدول look up است به یک خروجی ۴ بیتی تبدیل می‌کند $S\text{-box}$ ها قلب DES هستند و بدون آنها رمز خطی خواهد بود و در نتیجه قابل شکستن خواهد شد.

جایگشت: در نهایت ۳۲ بیت خروجی $S\text{-box}$ ها با استفاده از یک جایگشت ثابت مجدداً سازماندهی می‌شود ($P\text{-box}$).

الگوریتم Key Schedule

تولید کلید مراحل مختلف

از این الگوریتم برای تولید زیر کلیدها استفاده می‌شود. در ابتدا ۵۶ بیت از ۶۴ بیت کلید توسط انتخاب جایگشت ۱ ($PC1$) انتخاب می‌شوند و ۸ بیت باقی‌مانده یا دور ریخته می‌شوند و یا به عنوان parity برای چک کردن مورد استفاده قرار می‌گیرند سپس این ۵۶ بیت به دو نیمه ۲۸ تایی تقسیم می‌شوند و پس از آن با هرنیمه به طور مستقل رفتار

می‌شود. در دور بعدی هر دو نیمه یک یا دو بیت به سمت چپ انتقال می‌یابند.. سپس ۴۸ بیت زیرکلید توسط PC2 انتخاب می‌شوند. ۲۴ بیت، نیمه راست و ۲۴ بیت دیگر نیمه چپ را تشکیل می‌دهند. با استفاده از انتقال در هر زیر کلید مجموعه متفاوتی از بیتها مورد استفاده قرار می‌گیرد. هر بیت تقریباً در ۱۴ تا ۱۶ زیر کلید مورد استفاده واقع می‌شود. الگوریتم key schedule در رمزگشایی مانند رمزگذاری است ولی زیر کلیدها در مقایسه با رمزگذاری در جهت معکوس هستند به غیر از این تغییر، بقیه الگوریتم مانند رمزگذاری انجام می‌شود.

پیاده سازی در cryptool:

ابتدا در قسمت templates>>Cryptography>>Modern>>DES Cipher را باز می‌کنیم.

در قسمت Plaintext متن اصلی Sepide Omidvar را وارد می‌کنیم. در قسمت Key کلید را A1 B1 C1 11 11 11 در 11 11 11 در نظر می‌گیریم. قطعات را به صورت EBC می‌خواهیم رمز کنیم با padding صفرها. با زدن کلید start و تنظیم خروجی به صورت هگزادسیمال متن رمز شده 4B 77 55 1B 21 21 98 5E 52 6B C1 3B 5B B2 91 D8 خواهد بود.

