دانشکده مهندسی کامپیوتر

تمرین صفحه ۹۱ فصل ۲

امنیت سیستم‌های کامپیوتری

استاد درس: دکتر ابوالفضل دیانت

نام دانشجویان: فرزان رحمانی، محمدحسین عباسپور

شماره دانشجویی: ۹۹۵۲۱۲۷۱، ۹۹۵۲۱۴۳۳

نیم‌سال دوم

سال تحصیلی ۱۴۰۳-۱۴۰۲

پاسخ سوال اول

حمله تفاضلی یکی از حملات مهم به رمزگذاری داده‌ها است که اولین بار توسط بیهام و شامیر(Biham و Shamir) در طراحی الگوریتم DES معرفی شد. این حمله از روش تحلیل تفاضلی برای یافتن کلید رمزنگاری استفاده می‌کند. در این نوع حمله، دو بلوک متناظر با دو کلید متفاوت مقایسه می‌شوند تا تفاوت‌های موجود در اعداد دودویی آن‌ها بررسی شود. با تکرار این مرحله برای بلوک‌های مختلف، تفاوت‌هایی که با احتمال بالا در کلیدهای درست و غلط وجود دارند، شناسایی می‌شوند. سپس با انجام محاسبات مختلف و ترکیب این تفاوت‌ها، کلید استفاده شده در رمزنگاری پیدا می‌شود.

به عنوان مثال، برای یک الگوریتم DES با شش دور، ابتدا یک بلوک متنی با یک کلید تصادفی رمزگذاری می‌شود و خروجی آن با یک بلوک متنی دیگر رمزگذاری شده با کلید متفاوت مقایسه می‌شود. اگر تفاوت بین دو بلوک متنی در یک فاصله تفاضلی خاص باشد (مانند ۳ بیت)، آن تفاوت به عنوان نقطه شروع برای پیدا کردن کلید استفاده می‌شود. سپس با تکرار این فرآیند برای بلوک‌های دیگر، تفاوت‌هایی که با احتمال بالا در کلیدهای درست و غلط وجود دارند، شناسایی می‌شوند. در نهایت با ترکیب این تفاوت‌ها، کلید استفاده شده در رمزگذاری پیدا می‌شود.

تحلیل تفاضلی DES به این صورت است که تفاوت ورودی‌ها و خروجی‌ها در طی چندین دور از عملیات رمزنگاری پیگیری می‌شود. از آنجایی که DES از ساختار فایستل استفاده می‌کند، تحلیل تفاضلی با شناسایی الگوهای تکراری در توابع فایستل، امکان شناسایی کلید را فراهم می‌کند. به طور خاص، DES به دلیل طراحی ساده‌تر خود، در برابر این نوع حملات آسیب‌پذیر است. برای مثال، در تحلیل تفاضلی الگوریتم DES، تفاوت‌های ورودی در هر دور، تفاوت‌های خروجی خاصی را تولید می‌کنند که این تفاوت‌ها در نهایت به یافتن کلید منجر می‌شوند.

از آنجایی که حمله تفاضلی یک حمله احتمالاتی است، تعداد جفت‌های متن آشکار و رمز شده مورد نیاز برای انجام حمله به پیچیدگی الگوریتم رمزگذاری بستگی دارد. به طور کلی، با افزایش تعداد دورهای رمزگذاری، تعداد جفت‌های متن آشکار و رمز شده برای انجام حمله تفاضلی با موفقیت بیشتری مورد نیاز است. با این حال، حتی با یک نسخه ساده شده از الگوریتم DES با تعداد دورهای کم، حمله تفاضلی همچنان می‌تواند مؤثر باشد اگر ویژگی‌های تفاضلی با دقت انتخاب شوند.

حمله تفاضلی DES

در حمله تفاضلی به DES ، هدف اصلی این است که از تحلیل تفاوت‌های بین ورودی‌ها و خروجی‌های متن رمزنگاری شده، الگوهایی را استخراج کرده و کلید رمزنگاری را پیدا کنیم. این حمله به صورت عمومی شامل

مراحل زیر است:

۱. انتخاب متن آشکار: دو متن آشکار را انتخاب می‌کنیم که تفاوت مشخصی (مانند یک بیت) با هم داشته باشند.

۲. رمزنگاری متن‌ها: این دو متن آشکار را با همان کلید رمزنگاری می‌کنیم تا دو متن رمزنگاری شده به دست آید.

۳. تحلیل تفاضلی: تفاوت بین دو متن رمزنگاری شده را بررسی می‌کنیم و این تفاوت‌ها را با تفاوت‌های مورد انتظار مقایسه می‌کنیم.

۴. شناسایی کلید: با استفاده از تفاوت‌های مشاهده شده، کلید رمزنگاری را شناسایی می‌کنیم.

این فرایند با تکرار برای تعداد زیادی از جفت‌های متن آشکار و رمزنگاری شده انجام می‌شود تا الگوهای مورد نیاز برای شناسایی کلید پیدا شوند. در نهایت، با ترکیب این الگوها، کلید رمزنگاری به دست می‌آید.

به طور کلی، این الگوریتم در برابر حمله تفاضلی و این نوع تحلیل‌ها آسیب‌پذیر است و نیاز به طراحی الگوریتم‌های جدید و مقاوم‌تر در برابر حملات رمزنگاری را برجسته می‌کند.

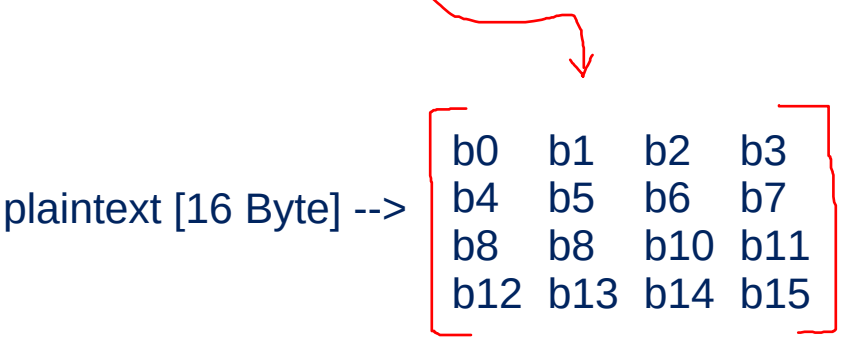
منبع: https://en.wikipedia.org/wiki/Differential_cryptanalysis

پاسخ سوال دوم

ابتدا با توضیح الگوریتم AES شروع می کنیم و سپس به مثال پایتون برای رمزگذاری و رمزگشایی یک پیام با استفاده از AES می‌رویم.

AES چگونه کار می کند؟

استاندارد رمزگذاری پیشرفته (AES) یک الگوریتم رمزگذاری متقارن است که توسط NIST استاندارد شده است. AES داده‌ها را در بلوک‌های ثابت ۱۲۸ بیتی (۱۶ بایتی) رمزگذاری می‌کند و از اندازه‌های کلیدی ۱۲۸، ۱۹۲ یا ۲۵۶ بیتی پشتیبانی می‌کند. در اینجا نکات کلیدی نحوه عملکرد AES آمده است:



مراحل رمزگذاری AES

۱. Expansion Key یا Schedule Key : کلید رمزگذاری به یک برنامه کلید بزرگتر گسترش می‌یابد. این کلید گسترش یافته از آرایه‌ای از کلمات کلیدی (هر کدام ۳۲ بیتی) تشکیل شده است و تعداد کل کلمات کلیدی به اندازه کلید بستگی دارد. برای AES-۱۲۸، Schedule Key شامل ۴۴ کلمه (۱۱ مجموعه ۴ کلمه‌ای)، برای AES-۱۹۲ دارای ۵۲ کلمه (۱۳ مجموعه ۴ کلمه‌ای) و برای AES-۲۵۶ دارای ۶۰ کلمه (۱۵ مجموعه ۴ کلمه‌ای) است.

۲. Round Initial :

- AddRoundKey : بلوک متن ساده با چهار کلمه اول جدول زمانی XOR می‌شود. این عملیات AddRoundKey نام دارد. هر بایت حالت را با یک بایت از کلید گرد با استفاده از XOR بیتی ترکیب می‌کند.

۳. Rounds Main : دورهای اصلی هر دور اصلی شامل چهار تبدیل است که به ترتیب اعمال می‌شوند:

- SubBytes (جایگزینی): مرحله جایگزینی غیر خطی که در آن هر بایت در بلوک با بایت دیگری با استفاده از یک جدول جایگزینی ثابت به نام S-box (جعبه جایگزینی) جایگزین می‌شود. این مرحله غیر خطی بودن رمز را فراهم می‌کند.
- ShiftRows (جایگشت): مرحله جابجایی که در آن ردیف‌های بلوک به صورت دوره‌ای جابجا می‌شوند. ردیف اول جابجا نمی‌شود، ردیف دوم با یک بایت به چپ، ردیف سوم با دو بایت و ردیف چهارم با سه بایت منتقل می‌شود. این مرحله انتشار بین ستونی را فراهم می‌کند.
- MixColumns (اختلاط): عملیات اختلاط که بر روی ستون‌های بلوک عمل می‌کند و داده‌ها را در هر ستون مخلوط می‌کند. هر بایت از یک ستون به یک مقدار جدید نگاشت می‌شود که تابعی از هر چهار بایت در ستون است. این مرحله انتشار در ستون‌ها را تضمین می‌کند.
- AddRoundKey: یکی دیگر از عملیات AddRoundKey که در آن بلوک با بخشی از کلید گسترش یافته XOR می‌شود.

substitution

۴. Round Final : دور نهایی شبیه به دور اصلی است اما مرحله MixColumns را حذف می‌کند. این شامل:

- SubBytes
- ShiftRows

- AddRoundKey

جزییات تبدیل‌ها

۱. SubBytes : از یک S-box برای انجام جایگزینی بایت به بایت بلوک استفاده می‌کند. S-box از معکوس ضربی بر روی $GF(2^8)$، همراه با یک تبدیل affine مشتق شده است.

۲. ShiftRows :شامل جابجایی ردیف‌های بلوک به سمت چپ است. مقدار جابجایی به شاخص ردیف بستگی دارد:

- ردیف ۰: بدون تغییر
- ردیف ۱: ۱ بایت به سمت چپ تغییر مکان دهید
- ردیف ۲: ۲ بایت به سمت چپ تغییر مکان دهید
- ردیف ۳: ۳ بایت به سمت چپ تغییر مکان دهید

۳. MixColumns : هر ستون بلوک را با ضرب آن در یک چند جمله‌ای ثابت تبدیل می‌کند. این ضرب در میدان محدود $GF(2^8)$ انجام می‌شود. تبدیل تضمین می‌کند که چهار بایت هر ستون با هم مخلوط شده و انتشار ستون را فراهم می‌کند.

۴. AddRoundKey : هر بایت بلوک با بایت مربوط به کلید گرد XOR می‌شود. این مرحله بلوک را با بخشی از زمانبندی کلید ترکیب می‌کند.

خلاصه

- Expansion Key : یک سری کلیدهای گرد را از کلید اولیه ایجاد می‌کند.
- Round Initial : کلید دور اولیه را به حالت اضافه می‌کند.
- Rounds Main : شامل مراحل SubBytes، ShiftRows MixColumns و AddRoundKey برای تبدیل مکرر حالت است.
- Round Final : شبیه دورهای اصلی است اما MixColumns را حذف می‌کند و فرآیند رمزگذاری را نهایی می‌کند.

نتیجه این مراحل متن رمزنگاری شده است که یک نسخه رمزگذاری شده ایمن از متن اصلی است. AES به دلیل امنیت و کارایی آن در اجرای سخت افزار و نرم افزار به طور گسترده‌ای مورد استفاده قرار می‌گیرد. برای

کاوش دقیق‌تر، می‌توانید به صفحه ویکی‌پدیا در AES مراجعه کنید.

مثال پایتون با استفاده از AES

نمونه‌ای از نحوه رمزگذاری و رمزگشایی یک پیام با استفاده از الگوریتم AES با کتابخانه pycryptodome در پایتون در نوت‌بوک Jupyter پیوست «Q۲_AES.ipynb» وجود دارد.

نصب و راه‌اندازی

ابتدا، اگر قبلاً نصب نکرده‌اید، باید کتابخانه pycryptodome را نصب کنید.

توضیح کد

۱. تولید کلید: ما از `get_random_bytes(16)` برای تولید یک کلید تصادفی ۱۶ بایتی برای رمزگذاری AES-۱۲۸ استفاده می‌کنیم.

۲. رمزگذاری:

- ما یک شی رمز AES جدید با `AES.new(key, AES.MODE_CBC)` ایجاد می‌کنیم.
- بردار مقداردهی اولیه (IV) به طور خودکار تولید شده و با `cipher.iv` بازیابی می‌شود.
- پیام با استفاده از pad، مضربی از اندازه بلوک (۱۶ بایت) است.
- پیام `padded` با استفاده از `cipher.encrypt` رمزگذاری می‌شود.
- IV برای استفاده در هنگام رمزگشایی با متن رمزی الحاق شده است.

۳. رمزگشایی:

- ما IV را از ۱۶ بایت اول پیام رمزگذاری شده استخراج می‌کنیم.
- ما یک شی رمز AES جدید برای رمزگشایی با همان کلید و IV ایجاد می‌کنیم.
- متن رمزی رمزگشایی شده و سپس با استفاده از `unpad` باز می‌شود.

این مثال رمزگذاری و رمزگشایی اولیه AES را با استفاده از حالت CBC (Chaining Block Cipher) نشان می‌دهد. برای اطلاعات بیشتر می‌توانید به صفحه Wikipedia AES برای توضیح دقیق اصول و طراحی الگوریتم مراجعه کنید.

منبع: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard