دانشکده مهندسی کامپیوتر

استاد درس: دکتر دیانت

بهار ۱۴۰۲

گزارش پروژه

درس امنیت سیستم های کامپیوتری

نیما کمبرانی، فاطمه زهرا بخشنده
شماره دانشجویی: ۹۸۵۲۱۴۲۳ ، ۹۸۵۲۲۱۵۷

مینی پروژه ۲

۱ سوال ۱

۱.۱ توضیحات

حمله تفاضلی در سال ۱۹۹۰ بر روی الگوریتم DES ارائه داده شد. این روش با این فرض کار می‌کند که با داشتن دو یا چند متن اولیه و متن رمز شده متناظر آن‌ها و بدست آوردن تفاضل هر یک از جفت متن‌ها و متن رمزشده متناظر آن‌ها ممکن است رابطه‌ای بین ریاضی بین تفاضل در متن اولیه و متن رمز وجود داشته باشد که با توجه به آن بتوان کل یا تعدادی از بیت های کلید را بدست آورد. درنتیجه با بدست آوردن برخی بیت های کلید شکستن کامل آن در زمان کمتری انجام خواهد شد و امنیت الگوریتم رمزنگاری از بین می‌رود. این روش می‌تواند با استفاده از روش های شکستن رمز متن رمز انتخابی (chosen cipher text attack) یا متن اولیه انتخابی (chosen plaintext attack) اجرا شوند.

در صورتی که الگوریتم نسبت به این نوع حمله مقاوم نباشد، می‌توان آن را با داشتن تعداد کمی متن ساده و متن رمز متناظر آن‌ها با سرعت بسیار بیشتر نسبت به Brute force اقدام به شکستن رمز کنیم. به عنوان مثال در الگوریتم FEAL-4 می‌توان تنها با داشتن ۸ جفت متن ساده انتخاب شده به سادگی الگوریتم را شکست.

برای انجام این حمله به صورت chosen plaintext ، ابتدا تعدادی جفت متن ورودی انتخاب می‌کنیم. متن های انتخاب شده باید یک اختلاف ثابت داشته باشند، مثلا همه آن‌ها در بیت ۵ ام اختلاف داشته باشند. پس از ساختن تعدادی جفت متن آشکار هر یک از متن‌ها با استفاده از کلید یکسان رمز می‌شوند. پس از رمزگذاری، اختلاف متن‌های رمز شده را نیز بدست می‌آوریم. پس از یافتن اختلاف متن‌های رمز شده با استفاده از تحلیل آماری به دنبال نشانه های از عدم وجود رابطه تصادفی در بین اختلاف متن رمز شده و متن اشکار می‌گردیم. بعنوان مثال اگر درصد زیادی از متن های رمز شده در یک بیت خاص تفاوت داشته باشند یا اختلاف یکسانی داشته باشند نشان از وجود رابطه بین تغییرات در متن اشکار و متن رمز شده است. در نتیجه با استفاده از این دانش می‌توان تعدادی از بیت های کلید را به دست آورد و عملیات شکستن کامل کلید را در زمان کمتری انجام داد.

همچنین برای انجام این حمله بصورت chosen ciphertext بصورت مشابه تعدادی متن رمز با اختلاف ثابت می‌سازیم و با بررسی روابط آماری در متن آشکار متناظر، به دنبال شواهد وجود الگو ها و روابط غیر تصادفی می‌گردیم.

مثال با DES سه دور: فرض کنید که می‌خواهیم یک پیام رمزنگاری شده با الگوریتم DES سه دوره رمزگشایی کنیم. اگر برای رمزنگاری این پیام از کلید 0123456789ABCDEF استفاده شود. سپس برای رمزنگاری پیام، از مراحل زیر عبور می‌کنیم:

مرحله ۱: اعمال تابع اولیه با ترکیب کلید و بلاک اول پیام

مرحله ۲: ۳ دور اعمال تابع‌های F و تابع جعبه جایگشتی

مرحله ۳: اعمال تابع اختصاصی با استفاده از کلید و بلاک آخر پیام

اگر پیام زیر را با کلید بالا رمزنگاری کنیم و پس از رمزنگاری، خروجی زیر را دریافت کنیم: 85E813540F0AB405

حال با فرض یک تفاوت یک بیتی بین ورودی‌های رمزنگاری، مثلاً بین پیام اصلی و پیامی که یک بیت آخر آن تغییر کرده است، با داشتن خروجی رمزنگاری این دو ورودی رمزنگاری را انجام داده و تفاوت خروجی را محاسبه می‌کنیم. اگر تفاوت خروجی برابر با یک الفبای ۱۶ بیتی باشد، احتمال دارد که برخی اجزای الگوریتم

رمزنگاری به یک شکل خاصی وابسته باشند و این ممکن است راهی برای برداشتن اطلاعات درباره الگوریتم باشد. این نوع حمله در طی چندین مرحله و با تلاش مکرر برای پیدا کردن تفاوت‌های دیگر، به محاسبه‌ی کلید استفاده شده در رمزنگاری پیام کمک می‌کند.

با توجه به شدت حملات تفاضلی بر روی lrDES/ ، استفاده از این الگوریتم به عنوان یک الگوریتم رمزنگاری امن، امروزه توصیه نمی‌شود و به جای آن، الگوریتم‌های قوی‌تر و امن‌تری مانند lrAES/ پیشنهاد می‌شوند.

۲ سوال ۲

۱.۲ الگوریتم AES

الگوریتم AES یک الگوریتم رمزنگاری متقارن از نوع Block cipher است که در بسیاری از کاربرد های نیازمند به رمز نگاری کاربرد دارد. این الگوریتم دارای طول کلید های ۱۲۸، ۱۹۶ و ۲۵۶ بیت برای است که هرچه طول کلید آن بلندتر باشد امنیت بالاتری دارد. این الگوریتم در هرمرتبه از ۴ عملگر اصلی برای رمز نگاری استفاده می‌کند. در ابتدا متن ورودی به بلاک های با طول ۱۲۸ بیت (۱۶ بایت) تقسیم می‌کند. سپس بلاک ۱۶ بایتی بدست آمده را به صورت یک ماتریس ۴*۴ در می‌آورد سپس مراحل رمز نگاری بر روی آن انجام می‌شود.

مراحل بصورت زیر هستند:

۱.۱.۲ مراحل

54	71	6b	6f
68	75	20	77
65	69	62	6e
20	63	72	20

شکل ۱: متن آشکار اولیه که یک بخش ۱۲۸ بیت از آن بصورت ماتریس ۴ در ۴ تبدیل شده است

54	71	6b	6f
68	75	20	77
65	69	62	6e
20	63	72	20

→

20	a3	7f	a8
45	9d	b7	f5
4d	f9	aa	9f
b7	fb	40	b7

شکل ۲: اعمال تابع غیرخطیsbox بر روی هر یک از بایت های متن

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

→

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,0}$
$b_{2,2}$	$b_{2,3}$	$b_{2,0}$	$b_{2,1}$
$b_{3,3}$	$b_{3,0}$	$b_{3,1}$	$b_{3,2}$

شکل ۳: جابه‌جا کردن و شیفت دادن ردیف ها

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \cdot \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

$$\begin{aligned} d_0 &= 2 \cdot b_o \oplus 3 \cdot b_1 \oplus 1 \cdot b_2 \oplus 1 \cdot b_3 \\ d_1 &= 1 \cdot b_o \oplus 2 \cdot b_1 \oplus 3 \cdot b_2 \oplus 1 \cdot b_3 \\ d_2 &= 1 \cdot b_o \oplus 1 \cdot b_1 \oplus 2 \cdot b_2 \oplus 3 \cdot b_3 \\ d_3 &= 3 \cdot b_o \oplus 1 \cdot b_1 \oplus 1 \cdot b_2 \oplus 2 \cdot b_3 \end{aligned}$$

شکل ۴: تبدیل ماتریس اعمال شده در هر مرحله بر روی هر ستون از متن

20	a3	7f	a8
9d	b7	f5	45
aa	9f	4d	f9
b7	b7	fb	40

→

e1	b7	4c	3d
53	db	a2	72
30	f3	06	c4
22	a3	d4	df

شکل ۵: اعمال تبدیل ماتریسی بر روی متن برای ترکیب کردن ستون ها

۲.۲ نمونه

۳.۲ کد سوال

کد سوال با استفاده از زبان پایتون پیاده شده، و از کتابخانه pycryptodome در آن استفاده است.
در این برنامه با گرفتن یک پیام از کاربر ابتدا آن را رمزنگاری کرده و مقادیر tag و nonce را نگه می‌داریم.
سپس در قسمت بعدی با استفاده از همین مقادیر، پیام رمزنگاری شده را رمزگشایی می کنیم.

e1	b7	4c	3d
53	db	a2	72
30	f3	06	c4
22	a3	d4	df

→

80	d2	25	50
31	bd	c8	1c
53	94	d6	ab
46	cb	b8	af

شکل ۶: ترکیب متن با کلید بصورت عملیات xor

```
D:\uni\security\projects\P02>python -u "d:\uni\security\projects\P02\AES_app.py"
Enter message to encrypt: hi everyone. this is a plaintext message.

Encrypted message:  b'\x04\xe3\xb7I\x01\x98\x95Lo\xa5\x1c\x8ab~\xcc\x94\xb2\n\x95\x1bf8,\xcb\x95\xf2\x8e\xc0\xb5\xde\
x95o\xdd\xd5x\x02\x80\xeab\r\xb9'
tag:  b'.\t\xd2\xc1\x83\xf3\xc0v\\n\x07\\{\xb6u\x91'
nonce value:  b'\x98\xdc\xb4\xae\xf9\x10\xde*\x1eBZ[\x1f\x0eD\xff'

Decrypted message:  hi everyone. this is a plaintext message.
```

شکل ۷: خروجی کنسول برای رمزنگاری و رمزگشایی یک پیام

در واقع، برای رمزگشایی یک پیام با استفاده از الگوریتم‌های رمزنگاری بلوکی اعتباردار مانند AES ، نیاز به داشتن هر دو مقدار nonce و tag است. nonce به عنوان یک مقدار تصادفی برای رمزنگاری از سمت فرستنده تولید می‌شود و به عنوان ورودی به الگوریتم رمزگشایی داده می‌شود. در هنگام رمزگذاری، tag توسط الگوریتم رمزگذاری تولید شده و در کنار متن رمزگذاری شده قرار می‌گیرد. در هنگام رمزگشایی، tag باید برای تأیید صحت متن رمزگذاری شده استفاده شود. بدون داشتن هر دو مقدار nonce و tag ، رمزگشایی پیام موفقیت آمیز نخواهد بود.

```python
from Crypto.Cipher import AES
import os

# Generate a random key
key = os.urandom(16)

# Encrypt the message
cipher = AES.new(key, AES.MODE_EAX)
message = input("Enter message to encrypt: ").encode()
ciphertext, tag = cipher.encrypt_and_digest(message)
nonce = cipher.nonce

print("\nEncrypted message: ", ciphertext)
print("tag: ", tag)
print("nonce value: ", nonce)

# Decrypt the message
cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)
decrypted = cipher.decrypt_and_verify(ciphertext, tag)

print("\nDecrypted message: ", decrypted.decode())
```

Listing :۱ encrypt and decrypt a message using AES