F21CN: Computer Network Security

Coursework 1 - Secret-Key Encryption

Andrew Beveridge
ab441@hw.ac.uk

11/10/2013

Contents

1 Lab Tasks **1**
1.1 Task 1: Encryption using different ciphers and modes 1
1.1.1 Shell Commands 1
1.2 Task 2: Encryption Mode — ECB vs. CBC 2
1.2.1 Shell Commands 2
1.3 Notes 4
1.3.1 Computing Power 4
1.3.2 OpenSSL 4
1.3.3 Editing Binary Files 4

List of Figures

1 Original Image 5
2 AES-ECB Encrypted Image 5
3 AES-CBC Encrypted Image 5
4 AES-OFB Encrypted Image 5

Listings

1 Task 1 1
2 Task 1 2

1 Lab Tasks

1.1 Task 1: Encryption using different ciphers and modes

My actions for this task can be summed up by the log of commands I ran, below. I was already quite familiar with OpenSSL so only compared the output from the different block cipher modes, directly on my web server using HexEdit.js

1.1.1 Shell Commands

Listing 1: Task 1

```
1  # pwd
2  /home/andrewbe/public_html/comnetsec
3  # ls -l
4  total 12
5  drwxr-xr-x 2 andrewbe andrewbe 4096 Oct 11 13:42 ./
6  drwxr-x--- 19 andrewbe nobody 4096 Oct 11 13:41 ../
7  -rw-r--r-- 1 andrewbe andrewbe 49 Oct 11 13:41 plain.txt
8  # openssl enc -aes-128-ecb -e -in plain.txt -out
       cipher-aes-128-ecb.bin -K 00112233445566778889aabbccddeeff -iv
       0102030405060708
9  # openssl enc -aes-128-cbc -e -in plain.txt -out
       cipher-aes-128-cbc.bin -K 00112233445566778889aabbccddeeff -iv
       0102030405060708
10 # openssl enc -aes-128-cfb -e -in plain.txt -out
       cipher-aes-128-cfb.bin -K 00112233445566778889aabbccddeeff -iv
       0102030405060708
11 # openssl enc -aes-128-cfb -e -in plain.txt -out
       cipher-aes-128-cfb.bin -K 00112233445566778889aabbccddeeff -iv
       0102030405060708
12 # openssl enc -aes-128-ofb -e -in plain.txt -out
       cipher-aes-128-ofb.bin -K 00112233445566778889aabbccddeeff -iv
       0102030405060708
13 # ls -l
14 total 28
15 drwxr-xr-x 2 andrewbe andrewbe 4096 Oct 11 13:42 ./
16 drwxr-x--- 19 andrewbe nobody 4096 Oct 11 13:41 ../
17 -rw-r--r-- 1 andrewbe andrewbe 64 Oct 11 13:42 cipher-aes-128-cbc.bin
18 -rw-r--r-- 1 andrewbe andrewbe 49 Oct 11 13:42 cipher-aes-128-cfb.bin
19 -rw-r--r-- 1 andrewbe andrewbe 64 Oct 11 13:42 cipher-aes-128-ecb.bin
20 -rw-r--r-- 1 andrewbe andrewbe 49 Oct 11 13:42 cipher-aes-128-ofb.bin
21 -rw-r--r-- 1 andrewbe andrewbe 49 Oct 11 13:41 plain.txt
22 #
```

1.2 Task 2: Encryption Mode — ECB vs. CBC

1.2.1 Shell Commands

Listing 2: Task 1

```
# wget http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/pic_original.bmp
--2013-10-11 14:04:33-- http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/pic_original.bmp
Resolving www.macs.hw.ac.uk... 137.195.13.48
Connecting to www.macs.hw.ac.uk|137.195.13.48|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 184974 (181K) [image/bmp]
Saving to: ‘pic_original.bmp’

100%[======================================================================>] 184,974 --.-K/s in 0.002s

2013-10-11 14:04:34 (99.1 MB/s) - ‘pic_original.bmp’ saved [184974/184974]
# ls -l
total 192
drwxr-xr-x 2 andrewbe andrewbe 4096 Oct 11 14:11 ./
drwxr-x--- 19 andrewbe nobody  4096 Oct 11 13:41 ../
-rw-r--r-- 1 andrewbe andrewbe 184974 Oct 11 14:04 pic_original.bmp
# openssl enc -aes-128-ecb -e -in pic_original.bmp -out pic-aes-128-ecb.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-cbc -e -in pic_original.bmp -out pic-aes-128-cbc.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-cfb -e -in pic_original.bmp -out pic-aes-128-cfb.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-ofb -e -in pic_original.bmp -out pic-aes-128-ofb.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-ofb -e -in pic_original.bmp -out pic-aes-128-ofb.bmp -K 00112233445566778889aabbccddeeff -iv 0102030405060708
# dd if=pic_original.bmp bs=1 count=54 | tee >(dd conv=notrunc of=pic-aes-128-ecb.bmp) >(dd conv=notrunc of=pic-aes-128-cbc.bmp) >(dd conv=notrunc of=pic-aes-128-cfb.bmp)| dd conv=notrunc of=pic-aes-128-ofb.bmp
54+0 records in
54+0 records out
54 bytes (54 B) copied, 4.7288e-05 s, 1.1 MB/s
0+1 records in
0+1 records out
54 bytes (54 B) copied, 2.1521e-05 s, 2.5 MB/s
0+1 records in
```

```
30          0+1 records out
31          54 bytes (54 B) copied, 0.000310489 s, 174 kB/s
32          0+1 records in
33          0+1 records out
34          54 bytes (54 B) copied, 0.000288515 s, 187 kB/s
35          0+12 records in
36          0+1 records out
37          54 bytes (54 B) copied, 0.000912583 s, 59.2 kB/s
38          #
```

As can be clearly seen from Figure 1 and Figure 2, Electronic codebook (ECB) should really not be used as a cipher as identical plain text blocks are encoded into the the same cipher blocks, meaning any patterns in the original data can be easily identified; it does not provide proper confidentiality of the plain text message.

1.3 Task 3: Encryption Mode — Corrupted Cipher Text

Prior to executing any commands, I hypothesized about what the results might show. I felt that CBC, OFB and CFB would have a lot of corruption in the decrypted file, as I thought a single byte becoming corrupt would have greater impact on the bytes around it. I felt ECB would be likely to only have one byte corrupt in the decrypted file, at the same offset as the corrupt byte in the encrypted file, due to the results with the image making me feel it encrypted each byte individually.

1.3.1 Shell Commands

Listing 3: Task 1

```
# nano morethan64bytes.txt
# openssl enc -aes-128-ofb -e -in morethan64bytes.txt -out
    morethan64bytes-aes-ofb.bin -K 00112233445566778889aabbccddeeff
    -iv 0102030405060708
# printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-ofb.bin
    bs=1 seek=30
1+0 records in
1+0 records out
1 byte (1 B) copied, 2.1001e-05 s, 47.6 kB/s
# openssl enc -aes-128-ofb -d -in morethan64bytes-aes-ofb.bin -out
    morethan64bytes-ofb-corrupt-decrypted.txt -K
    00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-cbc -e -in morethan64bytes.txt -out
    morethan64bytes-aes-cbc.bin -K 00112233445566778889aabbccddeeff
    -iv 0102030405060708
# printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-cbc.bin
    bs=1 seek=30
1+0 records in
1+0 records out
1 byte (1 B) copied, 1.7753e-05 s, 56.3 kB/s
# openssl enc -aes-128-cbc -d -in morethan64bytes-aes-cbc.bin -out
    morethan64bytes-cbc-corrupt-decrypted.txt -K
    00112233445566778889aabbccddeeff -iv 0102030405060708
# openssl enc -aes-128-ecb -e -in morethan64bytes.txt -out
    morethan64bytes-aes-ecb.bin -K 00112233445566778889aabbccddeeff
    -iv 0102030405060708
printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-ecb.bin bs=1
    seek=30
openssl enc -aes-128-ecb -d -in morethan64bytes-aes-ecb.bin -out
    morethan64bytes-ecb-corrupt-decrypted.txt -K
    00112233445566778889aabbccddeeff -iv 0102030405060708
# printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-ecb.bin
    bs=1 seek=30
1+0 records in
1+0 records out
1 byte (1 B) copied, 1.7091e-05 s, 58.5 kB/s
# openssl enc -aes-128-ecb -d -in morethan64bytes-aes-ecb.bin -out
    morethan64bytes-ecb-corrupt-decrypted.txt -K
```

```
    00112233445566778889aabbccddeeff -iv 0102030405060708
22  printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-cfb.bin bs=1
    seek=30
23  openssl enc -aes-128-cfb -d -in morethan64bytes-aes-cfb.bin -out
    morethan64bytes-cfb-corrupt-decrypted.txt -K
    00112233445566778889aabbccddeeff -iv 0102030405060708# openssl
    enc -aes-128-cfb -e -in morethan64bytes.txt -out morethancddeeff
    -iv 0102030405060708233445566778889aabbc
24  # printf '\x2F' | dd conv=notrunc of=morethan64bytes-aes-cfb.bin
    bs=1 seek=30
25  1+0 records in
26  1+0 records out
27  1 byte (1 B) copied, 1.5508e-05 s, 64.5 kB/s
28  # openssl enc -aes-128-cfb -d -in morethan64bytes-aes-cfb.bin -out
    morethan64bytes-cfb-corrupt-decrypted.txt -K
    00112233445566778889aabbccddeeff -iv 0102030405060708
29  # ls -la
30  total 44
31  drwxr-xr-x 2 andrewbe andrewbe 4096 Oct 11 15:11 ./
32  drwxr-x--- 19 andrewbe nobody 4096 Oct 11 13:41 ../
33  -rw-r--r-- 1 andrewbe andrewbe 368 Oct 11 15:11
    morethan64bytes-aes-cbc.bin
34  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:11
    morethan64bytes-aes-cfb.bin
35  -rw-r--r-- 1 andrewbe andrewbe 368 Oct 11 15:11
    morethan64bytes-aes-ecb.bin
36  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:09
    morethan64bytes-aes-ofb.bin
37  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:11
    morethan64bytes-cbc-corrupt-decrypted.txt
38  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:11
    morethan64bytes-cfb-corrupt-decrypted.txt
39  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:11
    morethan64bytes-ecb-corrupt-decrypted.txt
40  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:06
    morethan64bytes-ofb-corrupt-decrypted.txt
41  -rw-r--r-- 1 andrewbe andrewbe 359 Oct 11 15:03 morethan64bytes.txt
42  # cat morethan64bytes-cbc-corrupt-decrypted.txt
43  The job of ¡òwaxindKwxå}úWWdgently peeves c5intzy kids.
44  West quickly gave Bert handsome prizes for six juicy plums.
45  Just keep examining every low bid quoted for zinc etchings.
46  A quick movement of the enemy will jeopardize six gunboats.
47  All questions asked by five watch experts amazed the judge.
48  The exodus of jazzy pigeons is craved by squeamish walkers.
49  # cat morethan64bytes-cfb-corrupt-decrypted.txtÑÙô
50  Tob of waxing linoleum Éøfreu:
51     ntzy kids.
52  West quickly gave Bert handsome prizes for six juicy plums.
53  Just keep examining every low bid quoted for zinc etchings.
54  A quick movement of the enemy will jeopardize six gunboats.
55  All questions asked by five watch experts amazed the judge.
```

```
56          The exodus of jazzy pigeons is craved by squeamish walkers.
57          # cat morethan64bytes-ecb-corrupt-decrypted.txt
58          The job of ¡ªÚ¸ûwaxinmBa|Iently peeves chintzy kids.
59          West quickly gave Bert handsome prizes for six juicy plums.
60          Just keep examining every low bid quoted for zinc etchings.
61          A quick movement of the enemy will jeopardize six gunboats.
62          All questions asked by five watch experts amazed the judge.
63          The exodus of jazzy pigeons is craved by squeamish walkers.
64          # cat morethan64bytes-ofb-corrupt-decrypted.txt
65          The job of waxing linoleum freguently peeves chintzy kids.
66          West quickly gave Bert handsome prizes for six juicy plums.
67          Just keep examining every low bid quoted for zinc etchings.
68          A quick movement of the enemy will jeopardize six gunboats.
69          All questions asked by five watch experts amazed the judge.
70          The exodus of jazzy pigeons is craved by squeamish walkers.
71          #
```

My original hypothesis was incorrect, ECB, CBC and CFB all have about 13 bytes of corruption in the decrypted output, and OFB only has one corrupt byte, at the same offset.

1.4 Task 4: Checksums

1.4.1 Shell Commands

Listing 4: Task 1

```
# wget http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/some.aes-128-cbc
--2013-10-11 15:23:16-- http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/some.aes-128-cbc
Resolving www.macs.hw.ac.uk... 137.195.13.48
Connecting to www.macs.hw.ac.uk|137.195.13.48|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32 [text/plain]
Saving to: ‘some.aes-128-cbc’

100%[======================================================================>] 32 --.-K/s  in 0s

2013-10-11 15:23:16 (6.98 MB/s) - ‘some.aes-128-cbc’ saved [32/32]

# wget http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/some.txt
--2013-10-11 15:23:23-- http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/some.txt
Resolving www.macs.hw.ac.uk... 137.195.13.48
Connecting to www.macs.hw.ac.uk|137.195.13.48|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18 [text/plain]
Saving to: ‘some.txt’

100%[======================================================================>] 18 --.-K/s  in 0s

2013-10-11 15:23:24 (3.09 MB/s) - ‘some.txt’ saved [18/18]

# wget http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/words.txt
--2013-10-11 15:23:29-- http://www.macs.hw.ac.uk/~hwloidl/Courses/F21CN/Labs/CryptoI/words.txt
Resolving www.macs.hw.ac.uk... 137.195.13.48
Connecting to www.macs.hw.ac.uk|137.195.13.48|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 206662 (202K) [text/plain]
Saving to: ‘words.txt’

100%[======================================================================>] 206,662 --.-K/s in 0.04s

2013-10-11 15:23:29 (4.94 MB/s) - ‘words.txt’ saved [206662/206662]

# openssl sha1 some.aes-128-cbc
```

```
38          SHA1(some.aes-128-cbc)= 92ce63d9f3495ca005237eb6cca47302b74c574f
39          # openssl sha1 words.txt
40          SHA1(words.txt)= b3470280f84575a3db3ec3a6b9df2681ee0f5a18
41          # openssl sha1 some.txt
42          SHA1(some.txt)= 0b6f3556e8773a3e7c0ed31c634b9fd2a108adcc
43          #
```

1) The files have not been tampered with, as the hashes match with those published by the author. 2) Cryptographic hash functions 3) This check guarantees file /integrity/, as long as the hashes are checked against hashes which are known to be correct and sent over a secure channel.

1.5 Task 5: Known-plaintext attack

1.6 Notes

1.6.1 Computing Power

My first algorithm for task 5 was terribly inefficient, and took a long time (around 20 minutes) when run from linux01 (Core i7-860 CPU). I decided to try it on bwlf01 instead (Xeon E5504 CPU), as it seemed nobody was using that Beowulf node at the time. It took 12 minutes, a significant improvement, though still not great. I then ran it from my own dedicated server (Xeon E5-1650) and managed to squeeze the runtime down to only 6 minutes. At this point I realised the algorithm could be improved and tried another method, but I thought my computing power vs. time findings were worth a mention.

1.6.2 OpenSSL

I performed all the lab tasks from a virtual machine on my own dedicated server running a similar CentOS version as the MACS lab machines. As such, it had the same openssl-devel package installed as required.

1.6.3 Editing Binary Files

Some of the tasks in this coursework required me to directly edit and replace parts of binary files. Usually I would do this using a graphical hex editor such as HexEdit.js, but since I was working on a headless server I wanted to find a nicer way to replace portions of a file with another. After a little trial and error (and a brief reminder of the skip/seek flags) I ended up using dd, and found it was the perfect tool for the job, when you know your offsets as we did for task 2. This way, I avoided the awkward interface of shed, Emacs or (ugh) Vim for binary replacement!

Figure 1: Original Image

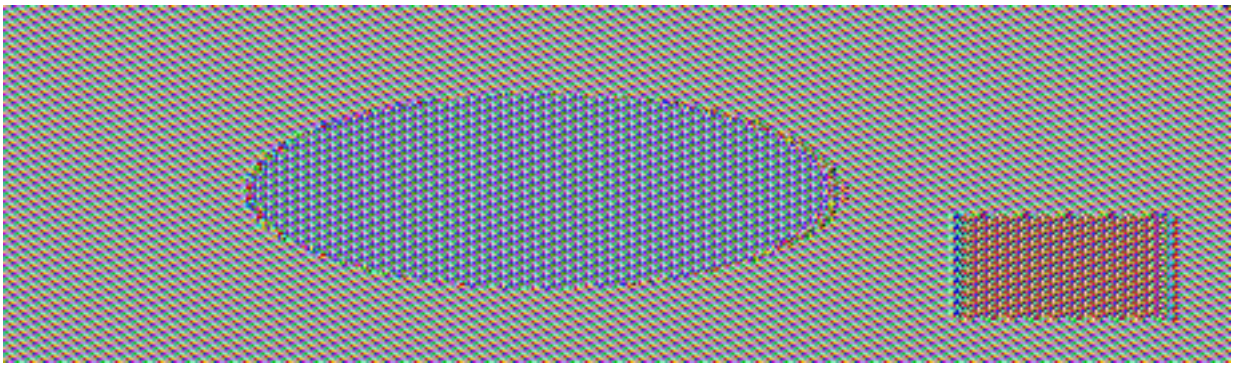

Figure 2: AES-ECB Encrypted Image

Figure 3: AES-CBC Encrypted Image

Figure 4: AES-OFB Encrypted Image