Course Title MCIS 6173
Term: Spring 2025

Laboratory Exercise 2 – Hands-on With Encryption Primitives and Block Cipher Modes of Operation

Due Date: Feb 21th, 2025, 11.59pm

1. Overview

This individual laboratory exercise will provide some hands-on experience with cryptography primitives and cipher block modes of operation on a Linux system.

2. Resources required

This exercise requires the Ubuntu with Snort and Other Tools exercise in the Virginia Cyber Range.

3. Initial Setup

Log in to your Virginia Cyber Range account and select the course **[Instructor: Information and Networking Security]** and the Exercise Environment "Ubuntu with Snort and Other Tools".

Click "start" to start your environment. When the systems are ready, click the "join" button and select the Primary Machine (desktop.example.com) from the dropdown menu to get to your Linux desktop login. If you are asked to log in, credentials for this Ubuntu Linux VM are below.

Username: **student**
Password: **student**

4. Tasks

Task 1: Lab Setup

First, open a Terminal window and switch to the lab3 directory on your Ubuntu virtual machine.

```
$ cd /home/student/lab3
```

Let's make sure we can encrypt and decrypt successfully using the **`openssl`** command-line tool, which provides a rich command set for using the various cryptographic functions from OpenSSL's crypto library. For a full list of openssl commands, see the man page:

```
$ man openssl
```

We'll be using the **`openssl enc`** command, which provides a tool for symmetric encryption and decryption using a variety of encryption algorithms and modes of operation.

For a description of the various command-line options for **`openssl enc`**, use the following command.

```
$ openssl enc --help
```

For a full list of algorithms and modes, use this command.

```
$ openssl enc -list
```

To encrypt the file **/home/student/lab3/test_message.txt** using 128-bit AES in Cypher-block Chaining (CBC) mode, use the syntax below. Here we are providing a passphrase (-k option) and initialization vector (-iv option).

```
$ openssl enc -aes-128-cbc -e –in test_message.txt –out
test_aes_128_cbc.bin \-k 00112233445566778899aabbccddeeff \-iv
0102030405060708
```

Attempt to view the contents of the file you created; you only see binary data. To test decryption, use the following syntax (be sure your passphrase and IV are identical to the above):

```
$ openssl enc -aes-128-cbc –d –in test_aes_128_cbc.bin –out
test_message2.txt \-k 00112233445566778899aabbccddeeff \-iv
0102030405060708
```

Now confirm that **test_message2.txt** is the same as **test_message.txt** (you can **cat** the 2 files or use **diff** for this step). If the two files are not identical, carefully examine your command history (you can use the Linux **history** utility for this) to be sure you typed them correctly.

Task 2: Comparing Block Cipher Modes of Operation

Comparing randomness of encryption output: We'll use the **openssl** utility to encrypt an image file using two different block cipher modes of operation and compare the results. Examine the file **/home/student/lab3/shapes.bmp** using the **Nomacs** image viewer software in your Cyber Range Virtual Machine (find it in the **Applications Menu** under **Graphics**). Close the file and use **openssl** to encrypt **shapes.bmp** using 128-bit AES in electronic codebook (ECB) mode and name the output file **shapes_aes-128-ecb.bmp**.

We now want to examine the resulting file in an image viewer. To do so, we have to fix the file's header using the **Okteta** hex editor so it is recognized as an image file. Use **Okteta** to copy the first 54 (0x36 hex) bytes from shapes.bmp and replace the first 84-bytes of shapes_aes-128-ecb.bmp (see figures below), then save the modified file.

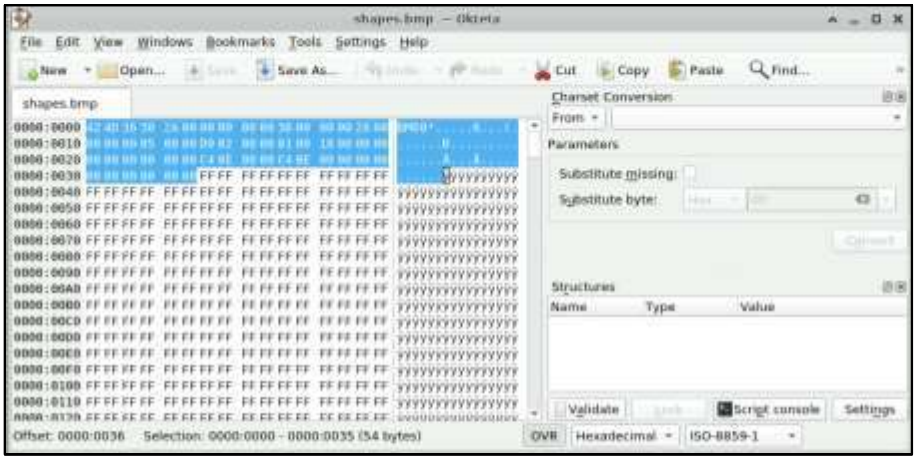


Figure 1. Use Okteta to copy the first 84 (hex 0x54) bytes from shapes.bmp.

Figure 2. shapes-aes-128-ecb.bmp before overwriting header data.

Figure 3. shapes-aes-128-ecb.bmp after overwriting header data.

Now open the file you just modified using **Nomacs** image viewer (Take a screenshot – you will be asked for it in the Canvas quiz).

1. What do you notice about the encrypted image file when viewed with the image viewer? Does it look 'random'? Why or why not? [Question 9 will ask for a screenshot.]

ANS: The encrypted image in ECB mode does not look fully random; it retains visible patterns from the original image. This happens because ECB encrypts identical plaintext blocks into identical ciphertext blocks, which allows some structure of the original image to be preserved.

Complete the above exercise, but this time use 128-bit AES with Cyber Block Chaining (CBC) mode to encrypt the file (and name the resulting file **shapes-aes-128-cbc.bmp**). Fix the header so that you can open the file in the Image Viewer (Take a screenshot – you will be asked for it in the Canvas quiz).

2. What do you notice about this encrypted file when viewed with the image viewer? Does it look 'random'? Why or why not? [Question 9 will ask for a screenshot.]

   ANS: The encrypted image in CBC mode appears much more randomized compared to ECB mode. This occurs because CBC mode uses an Initialization Vector (IV), ensuring that even identical plaintext blocks produce different ciphertext blocks. As a result, the image does not retain any visible patterns, making CBC more secure.

3. Based on this analysis, if you were trying to provide maximum protection to your encrypted files, which cipher block mode of operation would you use and why?

ANS: CBC mode is preferable to ECB because it eliminates patterns in the ciphertext. However, for even stronger security, Counter (CTR) mode or Galois/Counter Mode (GCM) is recommended as they provide both confidentiality and authentication.

Comparing robustness to bit errors: Files are sometimes corrupted in transit from sender to receiver, particularly over noisy wireless channels. We will next examine the effects of minor bit errors on encrypted data when the data are later decrypted.

Encrypt the file **plain.txt** using 128-bit AES in ECB mode. Use **Okteta** to increment the first character of the 40th (hex 0x28) byte (in the example below I change CA to DA, your values may be different) and save the file. Decrypt the file and examine the resulting plaintext.

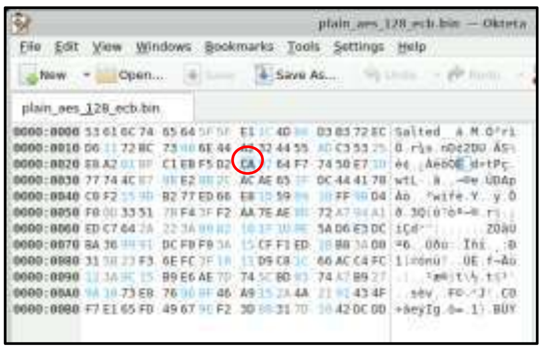


Figure 4. Encrypted file usign 128-bit AES in ECB mode before 40th byte is modified.

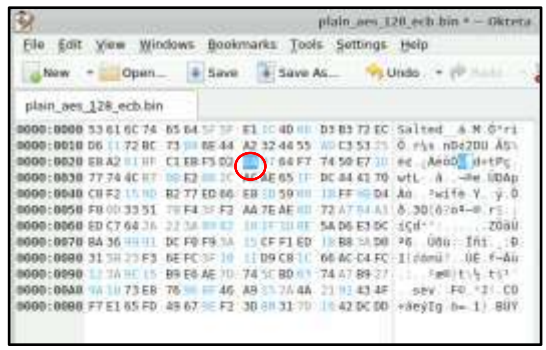


Figure 5. Encrypted file usign 128-bit AES in ECB mode after 40th byte is modified.

4. How badly was the file corrupted by modifying a single byte? Knowing what you should about ECB-mode, how many blocks were affected by your corruption of the ciphertext file?

ANS: A single-byte modification in the ciphertext affects only the corresponding byte in the decrypted plaintext.Number of affected blocks: 1 (only the corrupted byte's block).

Repeat the above exercise using CBC mode, Cipher Feedback (CFB) mode, and Output Feedback (OFB) mode, changing the first portion of the 40th byte of the encrypted file each time. Examine each decrypted file and answer the following questions.

5. CBC Mode: How badly was the file corrupted. Knowing what you know about CBC mode, how many blocks were affected by your corruption of the ciphertext file?

ANS: A single-byte modification in the ciphertext affects the entire corrupted block and the next block due to the chaining effect. Number of affected blocks: 2 (the modified block and the next one).

6. CFB Mode: How badly was the file corrupted. Knowing what you know about CFB mode, how many blocks were affected by your corruption of the ciphertext file?

ANS: A single-byte modification in the ciphertext affects only the corresponding byte in the decrypted plaintext.Number of affected blocks: 1 (only the affected byte).

7. OFB Mode: How badly was the file corrupted. Knowing what you know about OFB mode, how many blocks were affected by your corruption of the ciphertext file?

ANS: Similar to CFB mode, a single-byte modification affects only the corresponding byte in the decrypted plaintext.Number of affected blocks: 1 (only the affected byte).

8. Based on this exercise, if you were sending messages back and forth to a communicating partner on a noisy channel that suffers a lot of bit errors, which block mode of operation would you recommend using and why?

ANS: OFB Mode is the best choice, In ECB mode, encrypted images retain visible patterns since identical plaintext blocks produce identical ciphertext, making it insecure. CBC mode introduces an Initialization Vector (IV), ensuring better randomness and security by encrypting different identical blocks. In ECB mode, a single-byte modification affects only that byte, while in CBC mode, it corrupts both the modified block and the next one. CFB and OFB modes prevent error propagation, making OFB mode the best choice for secure transmission over noisy channels.

9. File upload – upload a .pdf or Word document with screenshots for questions 1 and 2. You may include any other screenshots you would like.

5. References

- Wikipedia has a good article on Block Cipher Modes of Operation at https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation.
- Dr. Mike Pound, at the University of Nottingham (https://www.nottingham.ac.uk/computerscience/) has a great YouTube video on Block Cipher Modes at https://www.youtube.com/watch?v=Rk0NIQfEXBA
- You can find lots of information about the OpenSSL community at https://www.openssl.org/

KSAs Addressed

From (http://csrc.nist.gov/publications/drafts/800-181/sp800_181_draft.pdf)

Knowledge:

- K0018: Knowledge of encryption algorithms (e.g., Internet Protocol Security [IPSEC], Advanced Encryption Standard [AES], Generic Routing Encapsulation [GRE], Internet Key Exchange [IKE], Message Digest Algorithm [MD5], Secure Hash Algorithm [SHA], Triple Data Encryption Standard [3DES]).
- K0049: Knowledge of information technology (IT) security principles and methods (e.g., firewalls, demilitarized zones, encryption).
- K0190: Knowledge of encryption methodologies.
- K0305: Knowledge of encryption algorithms, stenography, and other forms of data concealment.
- K0326: Knowledge of cybersecurity methods, such as firewalls, demilitarized zones, and encryption.
- K0417: Knowledge of data communications terminology (e.g., networking protocols, Ethernet, IP, encryption, optical devices, removable media).

- K0427: Knowledge of encryption algorithms and cyber capabilities/tools (e.g., SSL, PGP).
- K0439: Knowledge of governing authorities for targeting.
- K0561: Knowledge of the basics of network security (e.g., encryption, firewalls, authentication, honey pots, perimeter protection).

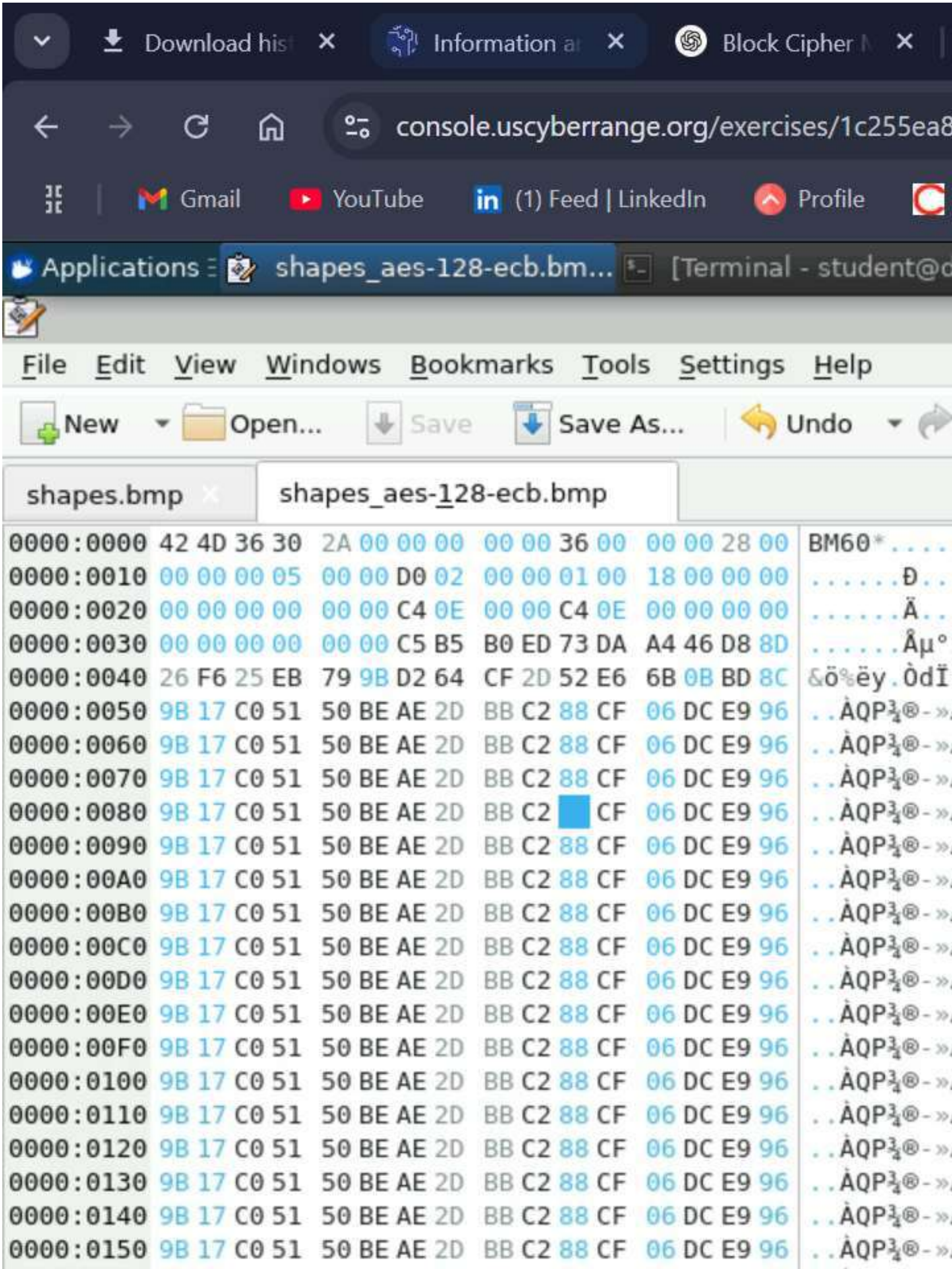
Skills:

- S0168: Skill in applying cybersecurity methods, such as firewalls, demilitarized zones, and encryption.

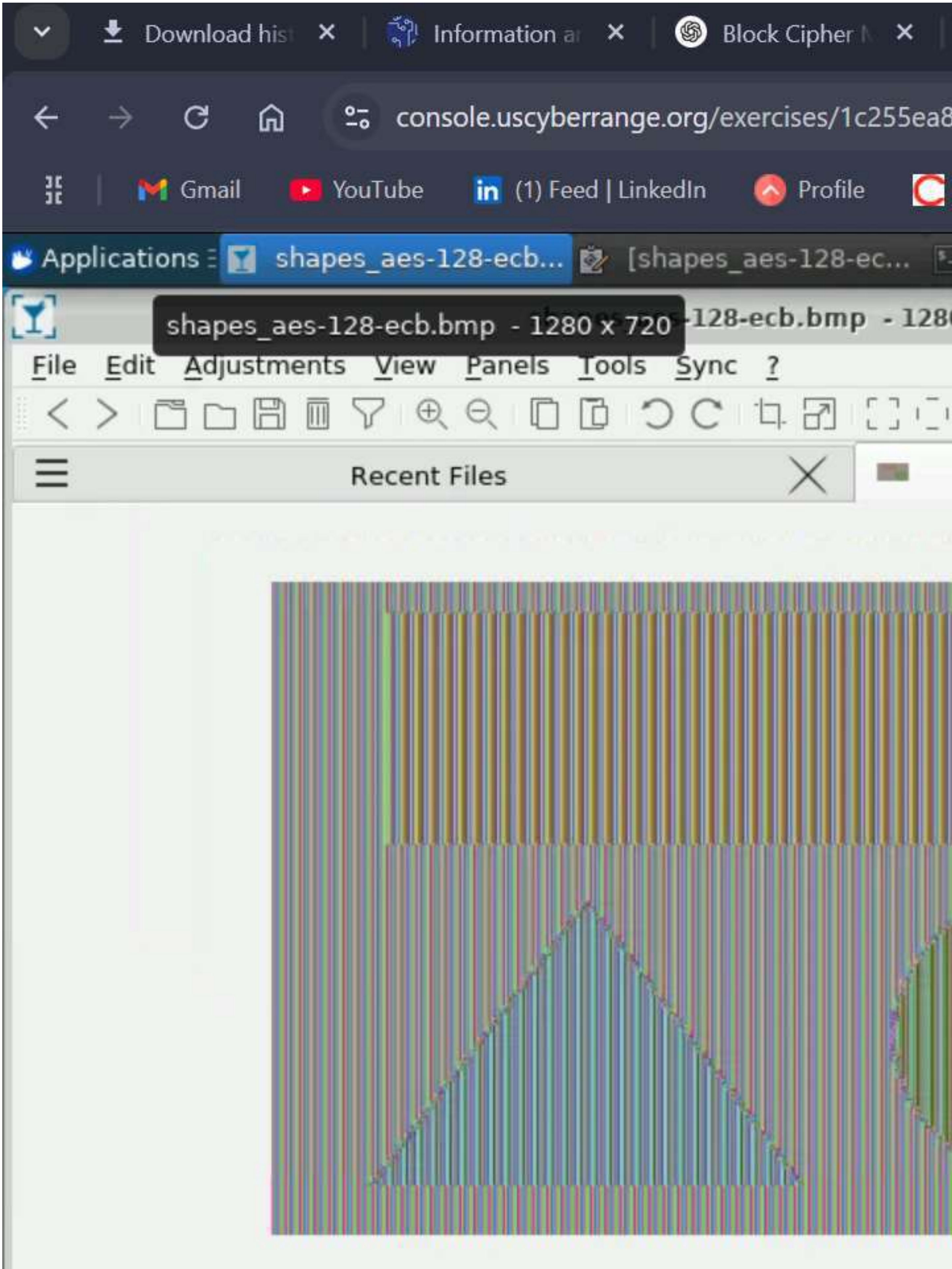
Knowledge Units (KUs) Addressed:

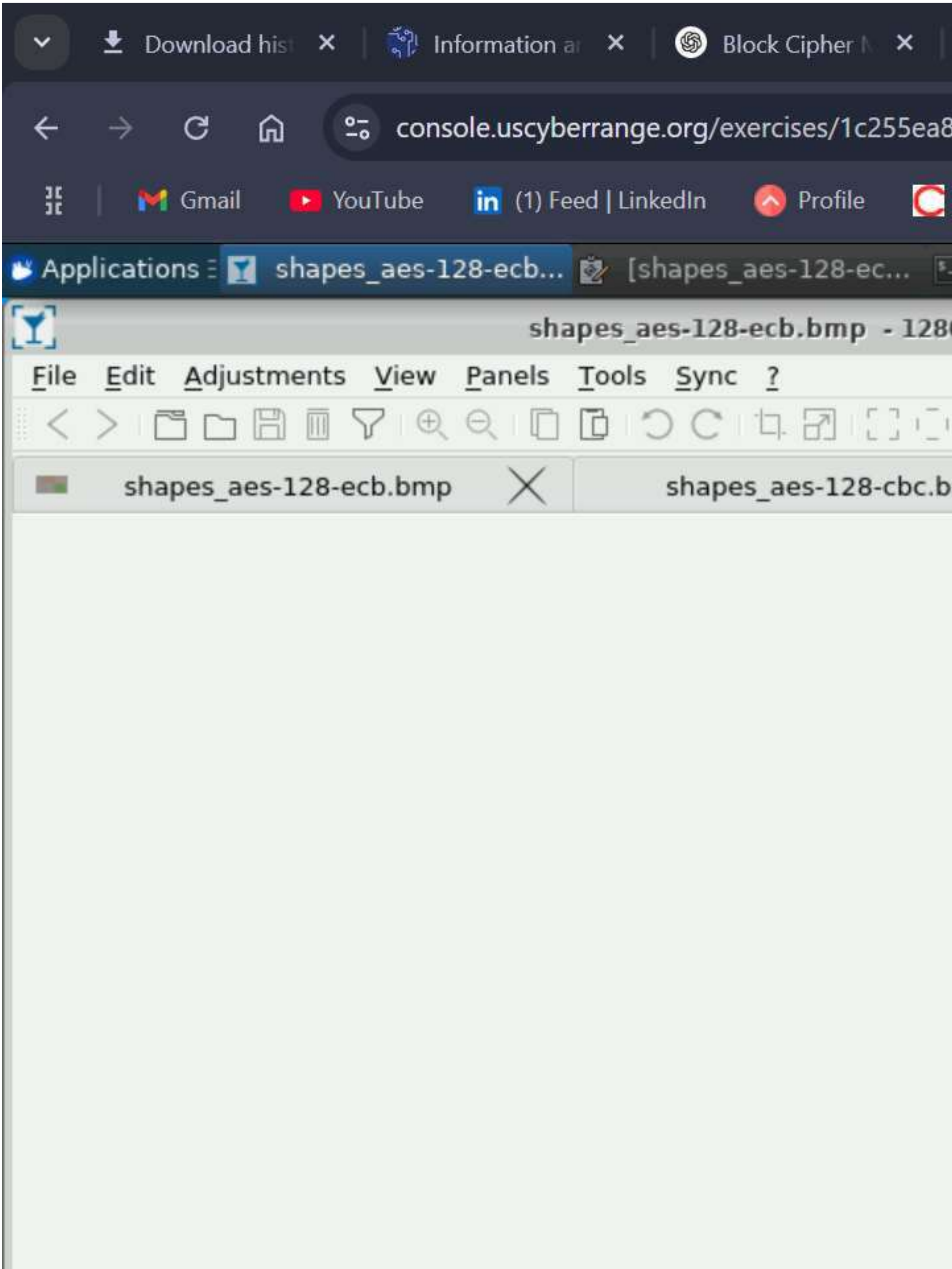
From (https://www.iad.gov/NIETP/documents/Requirements/CAE-CD_Knowledge_Units.pdf)

- IA Fundamentals
- Intro to Cryptography
- Advanced Cryptography (Modes and appropriate uses)

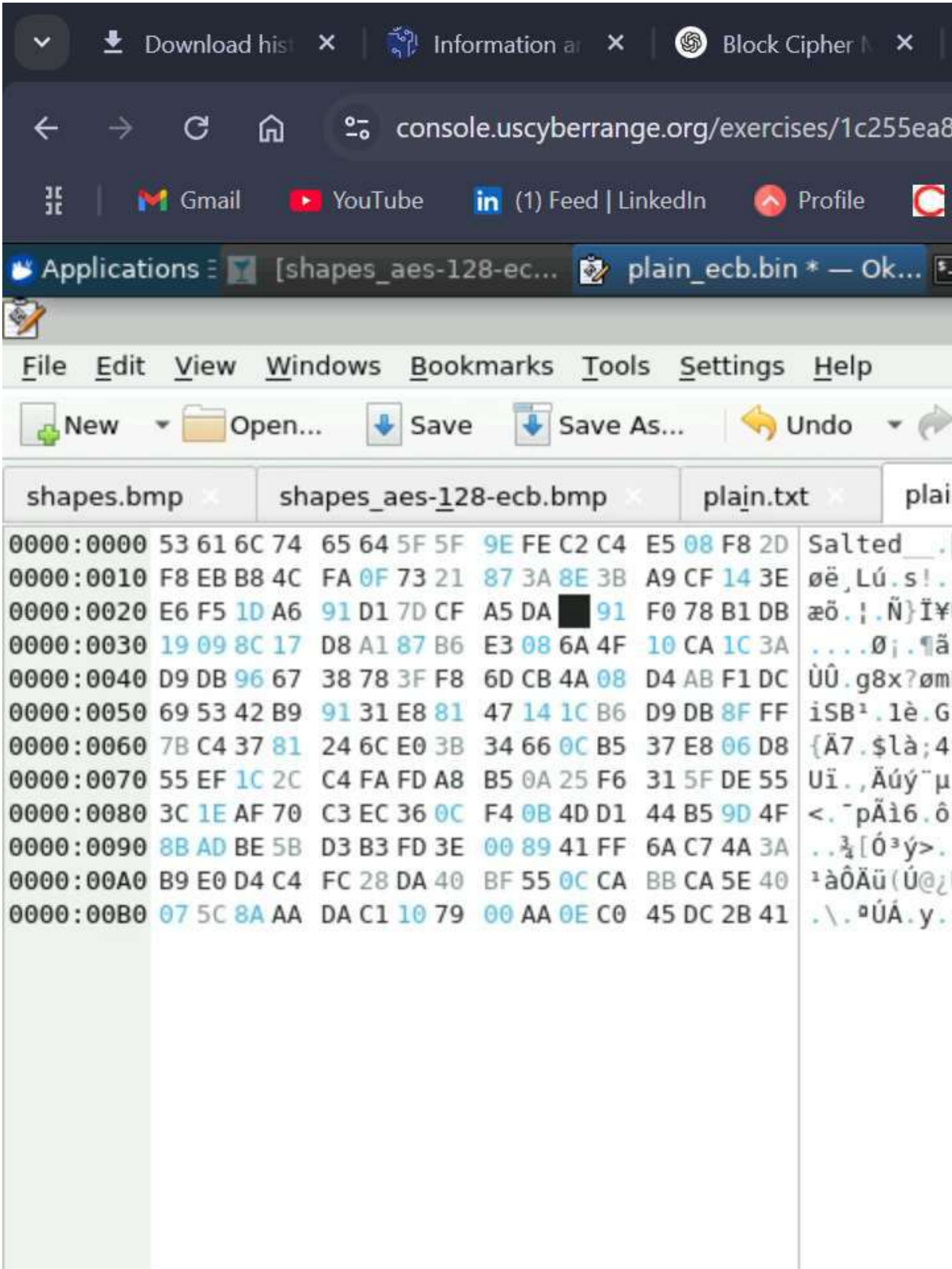


Download his
Information a
Block Cipher
console.uscyberrange.org/exercises/1c255ea8
Gmail
YouTube
(1) Feed | LinkedIn
Profile
Applications
shapes_aes-128-ecb.bm...
[Terminal - student@
File Edit View Windows Bookmarks Tools Settings Help
New
Open...
Save
Save As...
Undo
shapes.bmp
shapes_aes-128-ecb.bmp
0000:0000 42 4D 36 30 2A 00 00 00 00 00 36 00 00 00 28 00 BM60*.....
0000:0010 00 00 00 05 00 00 D0 02 00 00 01 00 18 00 00 00 ....Ð..
0000:0020 00 00 00 00 00 00 C4 0E 00 00 C4 0E 00 00 00 00 ....Ä..
0000:0030 00 00 00 00 00 00 C5 B5 B0 ED 73 DA A4 46 D8 8D ......Åµ°
0000:0040 26 F6 25 EB 79 9B D2 64 CF 2D 52 E6 6B 0B BD 8C &ö%ëy.ÒdÏ
0000:0050 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0060 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0070 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0080 9B 17 C0 51 50 BE AE 2D BB C2 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0090 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00A0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00B0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00C0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00D0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00E0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:00F0 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0100 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0110 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0120 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0130 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0140 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»
0000:0150 9B 17 C0 51 50 BE AE 2D BB C2 88 CF 06 DC E9 96 ..ÀQP¾®-»

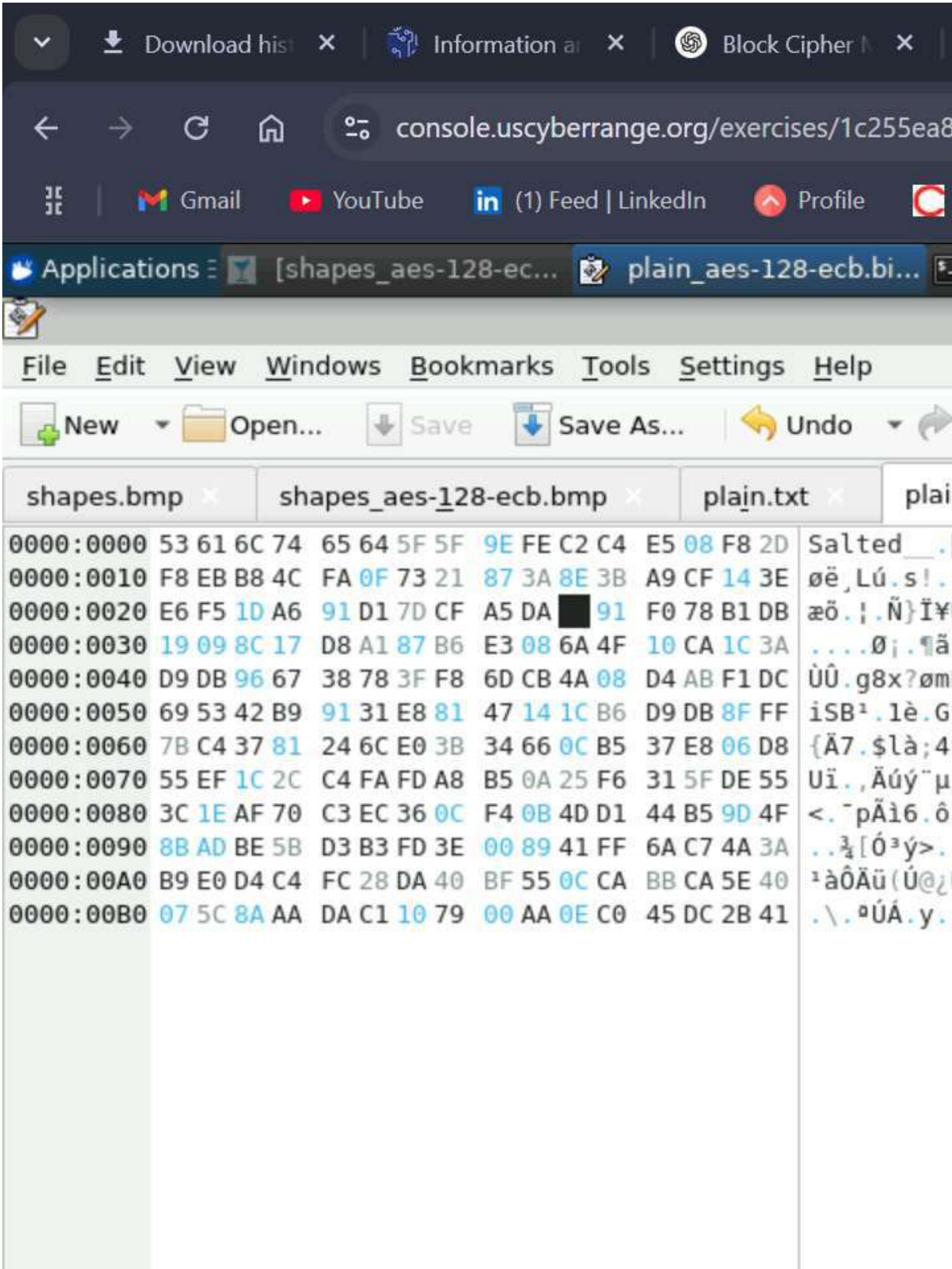

Download hist
Information ar
Block Cipher
console.uscyberrange.org/exercises/1c255ea8
Gmail
YouTube
(1) Feed | LinkedIn
Profile
Applications
shapes_aes-128-ecb...
[shapes_aes-128-ec...
shapes_aes-128-ecb.bmp - 1280 x 720
File
Edit
Adjustments
View
Panels
Tools
Sync
?
Recent Files



Download his
Information ar
Block Cipher
console.uscyberrange.org/exercises/1c255ea8
Gmail
YouTube
(1) Feed | LinkedIn
Profile
Applications
shapes_aes-128-ecb...
[shapes_aes-128-ec...
shapes_aes-128-ecb.bmp - 128
File
Edit
Adjustments
View
Panels
Tools
Sync
?
shapes_aes-128-ecb.bmp
shapes_aes-128-cbc.b



Download hist
Information ar
Block Cipher M
console.uscyberrange.org/exercises/1c255ea8
Gmail
YouTube
(1) Feed | LinkedIn
Profile
Applications
[shapes_aes-128-ec...
plain_ecb.bin * — Ok...
File Edit View Windows Bookmarks Tools Settings Help
New
Open...
Save
Save As...
Undo
shapes.bmp
shapes_aes-128-ecb.bmp
plain.txt
plai
0000:0000 53 61 6C 74 65 64 5F 5F 9E FE C2 C4 E5 08 F8 2D Salted__.
0000:0010 F8 EB B8 4C FA 0F 73 21 87 3A 8E 3B A9 CF 14 3E øë¸Lú.s!.
0000:0020 E6 F5 1D A6 91 D1 7D CF A5 DA [redacted] 91 F0 78 B1 DB æõ.¦.Ñ}Ï¥
0000:0030 19 09 8C 17 D8 A1 87 B6 E3 08 6A 4F 10 CA 1C 3A ....Ø¡.¶ã
0000:0040 D9 DB 96 67 38 78 3F F8 6D CB 4A 08 D4 AB F1 DC ÙÛ.g8x?øm
0000:0050 69 53 42 B9 91 31 E8 81 47 14 1C B6 D9 DB 8F FF iSB¹.1è.G
0000:0060 7B C4 37 81 24 6C E0 3B 34 66 0C B5 37 E8 06 D8 {Ä7.$là;4
0000:0070 55 EF 1C 2C C4 FA FD A8 B5 0A 25 F6 31 5F DE 55 Uï.,Äúý¨µ
0000:0080 3C 1E AF 70 C3 EC 36 0C F4 0B 4D D1 44 B5 9D 4F <.¯pÃì6.ô
0000:0090 8B AD BE 5B D3 B3 FD 3E 00 89 41 FF 6A C7 4A 3A ..¾[Ó³ý>.
0000:00A0 B9 E0 D4 C4 FC 28 DA 40 BF 55 0C CA BB CA 5E 40 ¹àÔÄü(Ú@¿
0000:00B0 07 5C 8A AA DA C1 10 79 00 AA 0E C0 45 DC 2B 41 .\.ªÚÁ.y.


Download hist
Information ar
Block Cipher
console.uscyberrange.org/exercises/1c255ea8
Gmail
YouTube
(1) Feed | LinkedIn
Profile
Applications
[shapes_aes-128-ec...
plain_aes-128-ecb.bi...
File Edit View Windows Bookmarks Tools Settings Help
New
Open...
Save
Save As...
Undo
shapes.bmp
shapes_aes-128-ecb.bmp
plain.txt
plai
0000:0000 53 61 6C 74 65 64 5F 5F 9E FE C2 C4 E5 08 F8 2D Salted__.
0000:0010 F8 EB B8 4C FA 0F 73 21 87 3A 8E 3B A9 CF 14 3E øë¸Lú.s!.
0000:0020 E6 F5 1D A6 91 D1 7D CF A5 DA [redacted] 91 F0 78 B1 DB æõ.¦.Ñ}Ï¥
0000:0030 19 09 8C 17 D8 A1 87 B6 E3 08 6A 4F 10 CA 1C 3A ....Ø¡.¶ã
0000:0040 D9 DB 96 67 38 78 3F F8 6D CB 4A 08 D4 AB F1 DC ÙÛ.g8x?øm
0000:0050 69 53 42 B9 91 31 E8 81 47 14 1C B6 D9 DB 8F FF iSB¹.1è.G
0000:0060 7B C4 37 81 24 6C E0 3B 34 66 0C B5 37 E8 06 D8 {Ä7.$là;4
0000:0070 55 EF 1C 2C C4 FA FD A8 B5 0A 25 F6 31 5F DE 55 Uï.,Äúý¨µ
0000:0080 3C 1E AF 70 C3 EC 36 0C F4 0B 4D D1 44 B5 9D 4F <.¯pÃì6.ô
0000:0090 8B AD BE 5B D3 B3 FD 3E 00 89 41 FF 6A C7 4A 3A ..¾[Ó³ý>.
0000:00A0 B9 E0 D4 C4 FC 28 DA 40 BF 55 0C CA BB CA 5E 40 ¹àÔÄü(Ú@¿
0000:00B0 07 5C 8A AA DA C1 10 79 00 AA 0E C0 45 DC 2B 41 .\.ªÚÁ.y.