



Faculty of Engineering and Technology  
Electrical and Computer Engineering Department  
**ENCS4320-APPLIED CRYPTOGRAPHY**  
**Homework #2 Test Cases**

**Tiny Encryption Algorithm -TEA- with:  
Electronic Code Book Mode-ECB-  
Cipher Block Chaining Mode-CBC-**

Name:	<b>Qusay Taradeh</b>
ID No. :	<b>1212508</b>
Instructor:	<b>Dr.Ahmad Shawahna</b>
Section No. :	<b>1</b>

# 1. Electronic Code Book (ECB) Mode:

## 1.1. Text Encryption/Decryption:

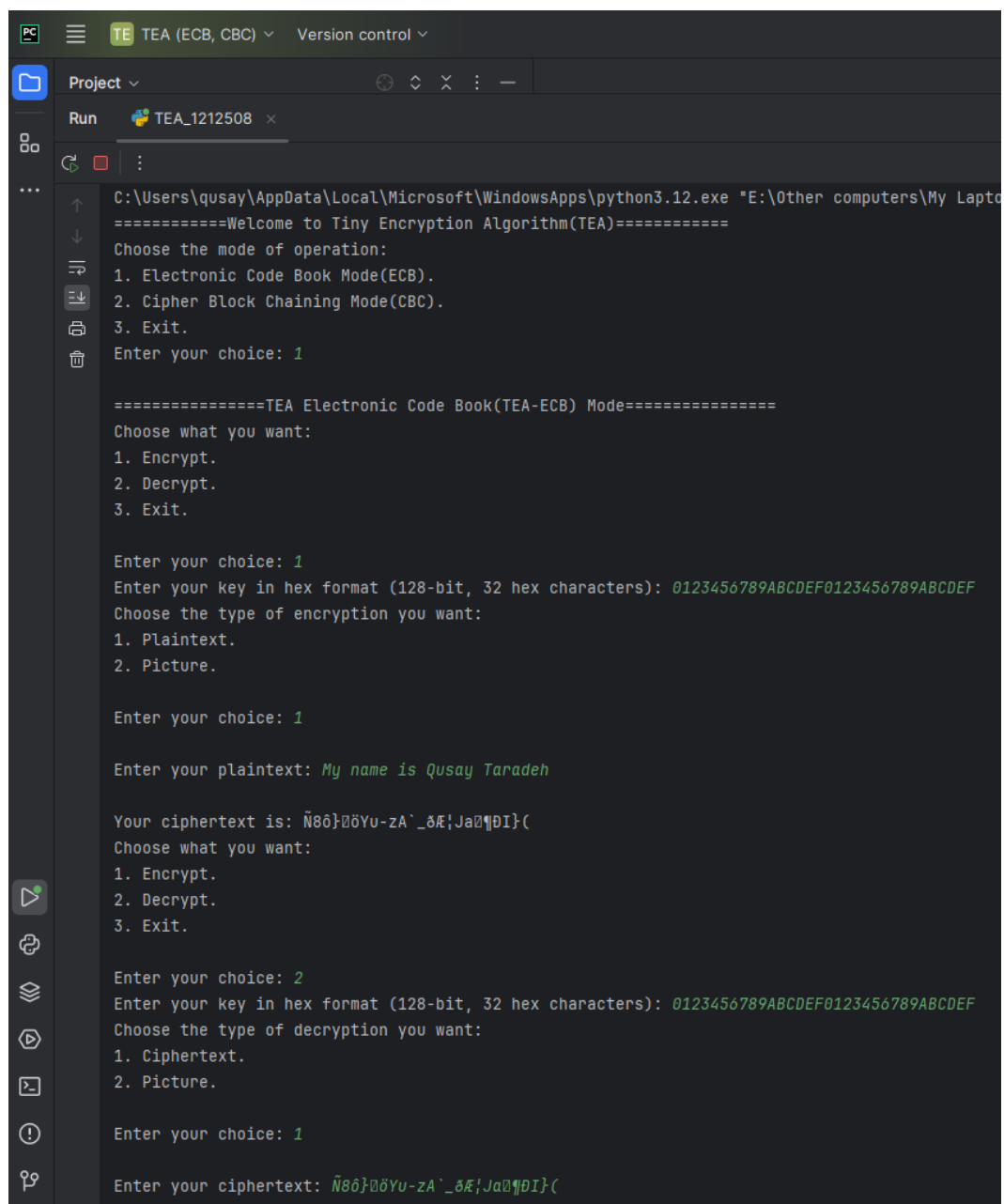
**Encrypts/Decrypts each 64-bit block of plaintext/ciphertext individually using 128-bit key as following:**

**Plaintext:** “My name is Qusay Taradeh”

**Key:** (0123456789ABCDEF0123456789ABCDEF)<sub>16</sub>

**Ciphertext:** “Ñ8ô}%öYu-zA`\_ðÆ!Ja ¶ÐI}{”

As you see in the Figure 1-1 and Figure 1-2 below:



```
PC TEA (ECB, CBC) Version control
Project
Run TEA_1212508 x
C:\Users\qusay\AppData\Local\Microsoft\WindowsApps\python3.12.exe "E:\Other computers\My Laptop"
=====Welcome to Tiny Encryption Algorithm(TEA)=====
Choose the mode of operation:
1. Electronic Code Book Mode(ECB).
2. Cipher Block Chaining Mode(CBC).
3. Exit.
Enter your choice: 1

=====TEA Electronic Code Book(TEA-ECB) Mode=====
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 1
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of encryption you want:
1. Plaintext.
2. Picture.

Enter your choice: 1
Enter your plaintext: My name is Qusay Taradeh

Your ciphertext is: Ñ8ô}%öYu-zA`_ðÆ!Ja¶ÐI}{
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 1
Enter your ciphertext: Ñ8ô}%öYu-zA`_ðÆ!Ja¶ÐI}{
```

Figure 1-1 ECB-Text Enc/Dec

```

Your ciphertext is: N8ô}0ôYU-zA`_ðÆ!Ja0¶0I}(
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 1

Enter your ciphertext: N8ô}0ôYU-zA`_ðÆ!Ja0¶0I}(

Your plaintext is: my name is qusay taradeh
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 3

Exit from ECB mode...

Choose the mode of operation:
1. Electronic Code Book Mode(ECB).
2. Cipher Block Chaining Mode(CBC).
3. Exit.
Enter your choice: 2

=====TEA Cipher Block Chaining(TEA-ECB) Mode=====

```

Figure 1-2 ECB-Text Enc/Dec

## 1.2. Picture Encryption/Decryption::

**Encrypts/Decrypts each 64-bit block of picture bytes individually using 128-bit key as following:**

**Key:** (0123456789ABCDEF0123456789ABCDEF)<sub>16</sub>

**Picture Path:** “Aqsa.bmp”

As you see in Figure 1-3 application below, and Figures 1-5 and 1-6 encrypted picture and decrypted one:

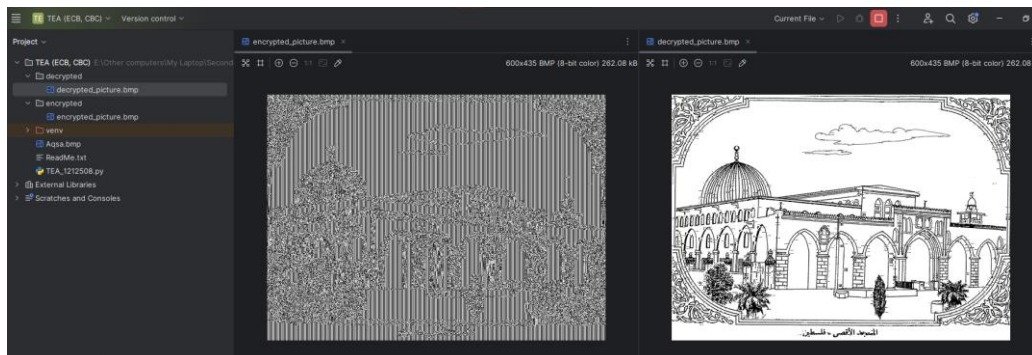


Figure 1-3 ECB-Picture Enc/Dec

```

Exit from CBC mode...

Choose the mode of operation:
1. Electronic Code Book Mode(ECB).
2. Cipher Block Chaining Mode(CBC).
3. Exit.
Enter your choice: 1

=====TEA Electronic Code Book(TEA-ECB) Mode=====
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 1
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of encryption you want:
1. Plaintext.
2. Picture.

Enter your choice: 2
Enter your picture path:Aqsa.bmp

Encrypted picture stored in folder called encrypted
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 2
Enter your picture path:encrypted/encrypted_picture.bmp

Decrypted picture stored in folder called decrypted

```

Figure 1-4 ECB-Picture Enc/Dec

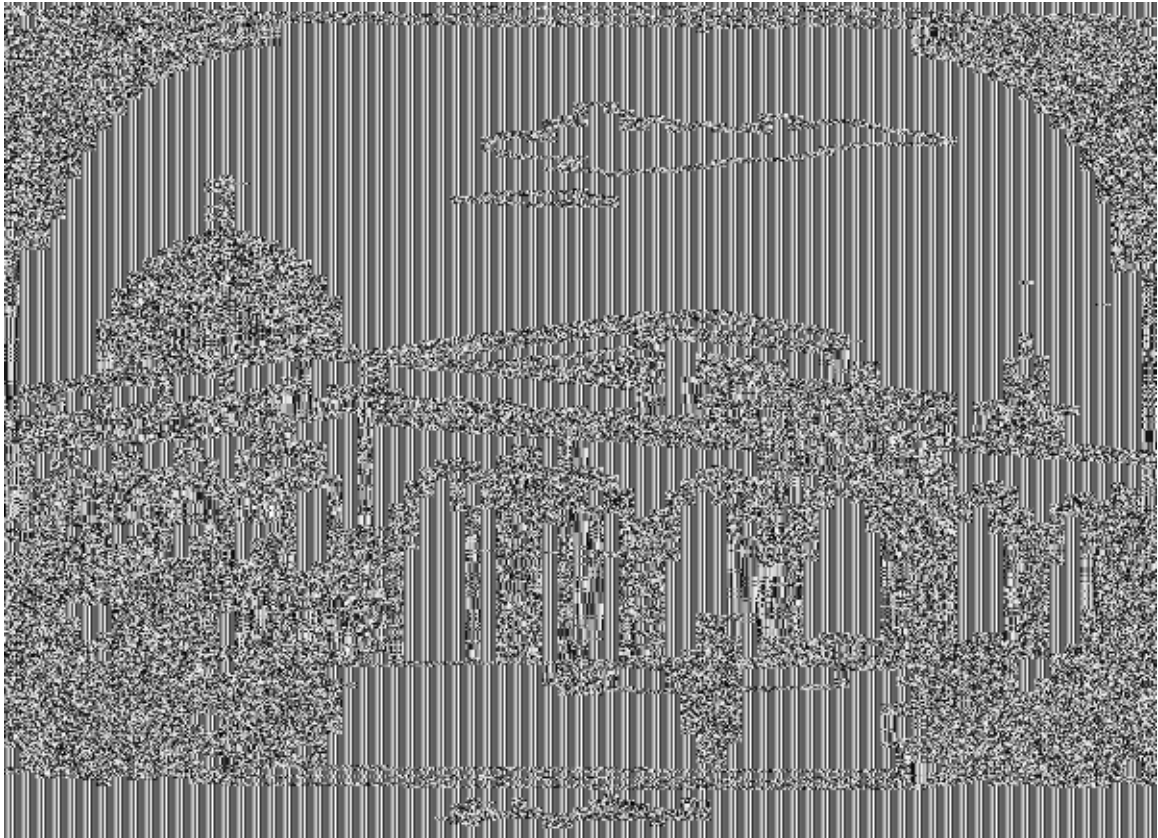


Figure 1-5 ECB-Picture Encryption

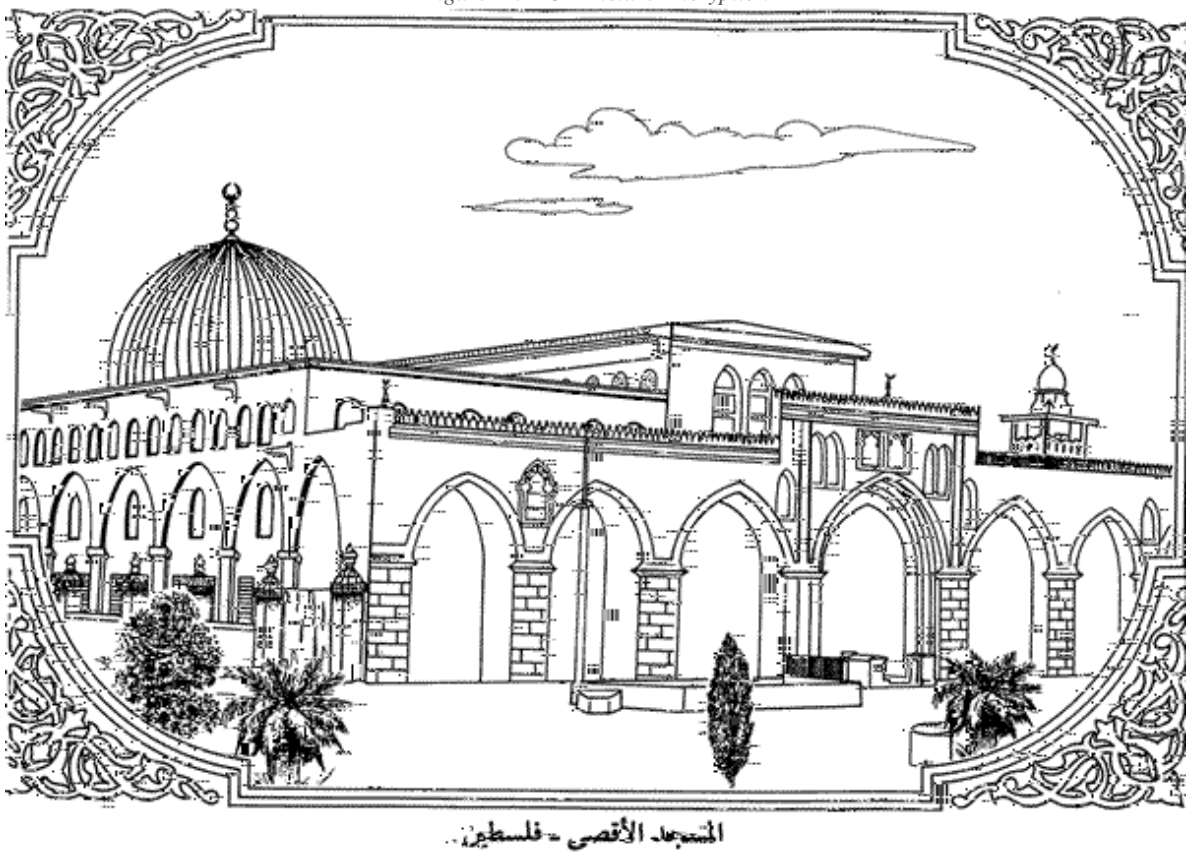


Figure 1-6 ECB-Picture Decryption

## 2. Cipher Block Chaining (CBC) Mode:

### 2.1. Text Encryption/Decryption:

**Each Block of ciphertext depends on previous cipher except first one equals Initialization Vector (IV), Also for Decryption but the difference is the IV included in Ciphertext.**

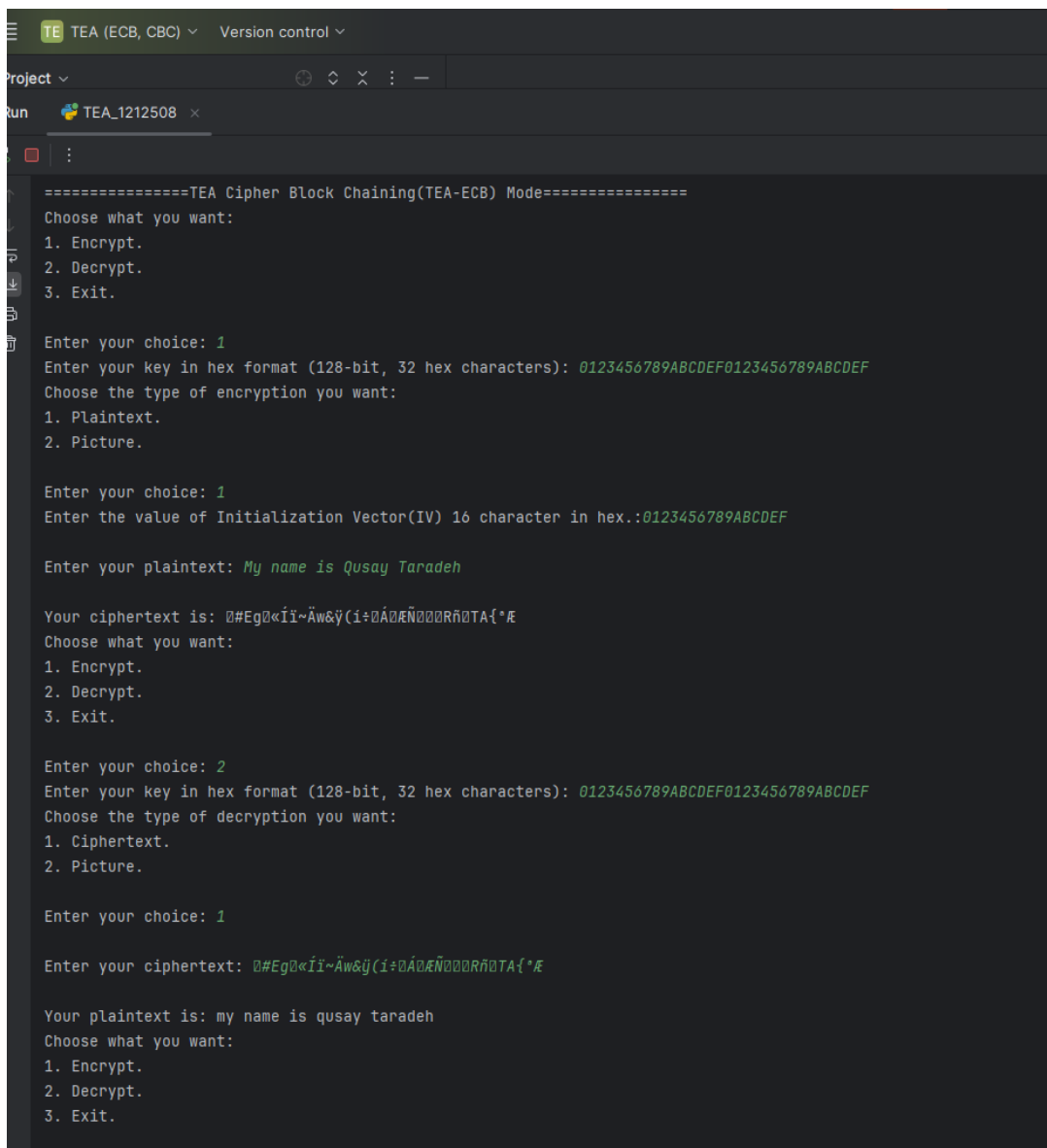
**Plaintext:** “My name is Qusay Taradeh”

**Key:** (0123456789ABCDEF0123456789ABCDEF)<sub>16</sub>

**IV:** (0123456789ABCDEF)<sub>16</sub>

**Ciphertext:** “ #Eg%«Íi~Äw&ÿ(i÷Š Á ÆÑ ™ Rñ• TA{ªÆ”

As in Figure 2-7 below



```
TEA (ECB, CBC) Version control
Project
Run TEA_1212508
=====TEA Cipher Block Chaining(TEA-ECB) Mode=====
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 1
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of encryption you want:
1. Plaintext.
2. Picture.

Enter your choice: 1
Enter the value of Initialization Vector(IV) 16 character in hex.:0123456789ABCDEF

Enter your plaintext: My name is Qusay Taradeh

Your ciphertext is: #Eg%«Íi~Äw&ÿ(i÷Š Á ÆÑ ™ Rñ• TA{ªÆ
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 1
Enter your ciphertext: #Eg%«Íi~Äw&ÿ(i÷Š Á ÆÑ ™ Rñ• TA{ªÆ

Your plaintext is: my name is qusay taradeh
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.
```

Figure 2-1 CBC Text Enc/Dec



## 2.2. Picture Encryption/Decryption:

Also, in pictures but here in CBC also need IV to encrypt pictures.

**Key:** (0123456789ABCDEF0123456789ABCDEF)<sub>16</sub>

**IV:** (0123456789ABCDEF)<sub>16</sub>

**Picture Path:** “Aqsa.bmp”

As you see in Figures 2-2, 2-3 and 2-4 application below, and Figures 2-5 and 2-6 encrypted picture and decrypted one:

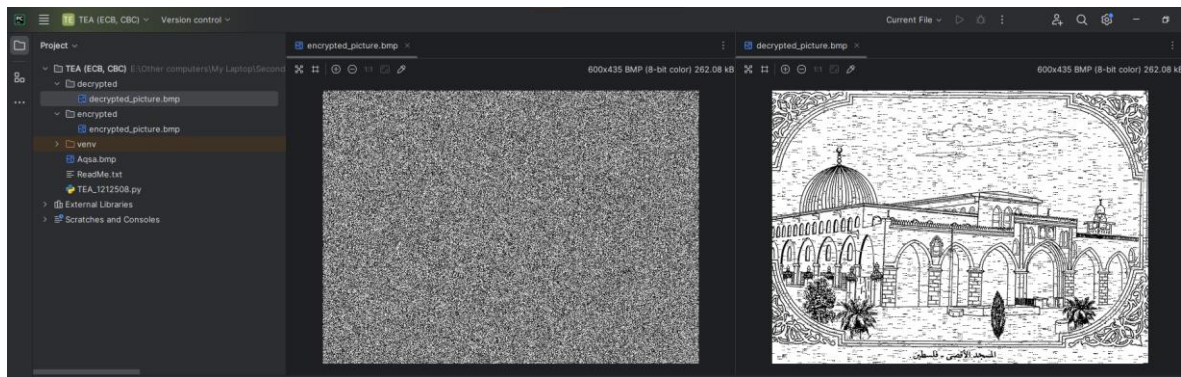


Figure 2-2 CBC Picture Enc/Dec

```

Choose the mode of operation:
1. Electronic Code Book Mode(ECB).
2. Cipher Block Chaining Mode(CBC).
3. Exit.
Enter your choice: 2

=====TEA Cipher Block Chaining(TEA-ECB) Mode=====
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 1
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of encryption you want:
1. Plaintext.
2. Picture.

Enter your choice: 2
Enter your picture path:Aqsa.bmp
Enter the value of Initialization Vector(IV) 16 character in hex.:0123456789ABCDEF

Encrypted picture stored in folder called encrypted
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 2
Enter your picture path:encrypted/encrypted_picture.bmp

```

Figure 2-3 CBC Picture Enc/Dec

```

Enter your choice: 2
Enter your key in hex format (128-bit, 32 hex characters): 0123456789ABCDEF0123456789ABCDEF
Choose the type of decryption you want:
1. Ciphertext.
2. Picture.

Enter your choice: 2
Enter your picture path:encrypted/encrypted_picture.bmp

Decrypted picture stored in folder called decrypted
Choose what you want:
1. Encrypt.
2. Decrypt.
3. Exit.

Enter your choice: 3

Exit from CBC mode...

Choose the mode of operation:
1. Electronic Code Book Mode(ECB).
2. Cipher Block Chaining Mode(CBC).
3. Exit.
Enter your choice: 3
=====Thank you for using Tiny Encryption Algorithm(TEA)!=====

Process finished with exit code 0
|

```

Figure 2-4 CBC Picture Enc/Dec



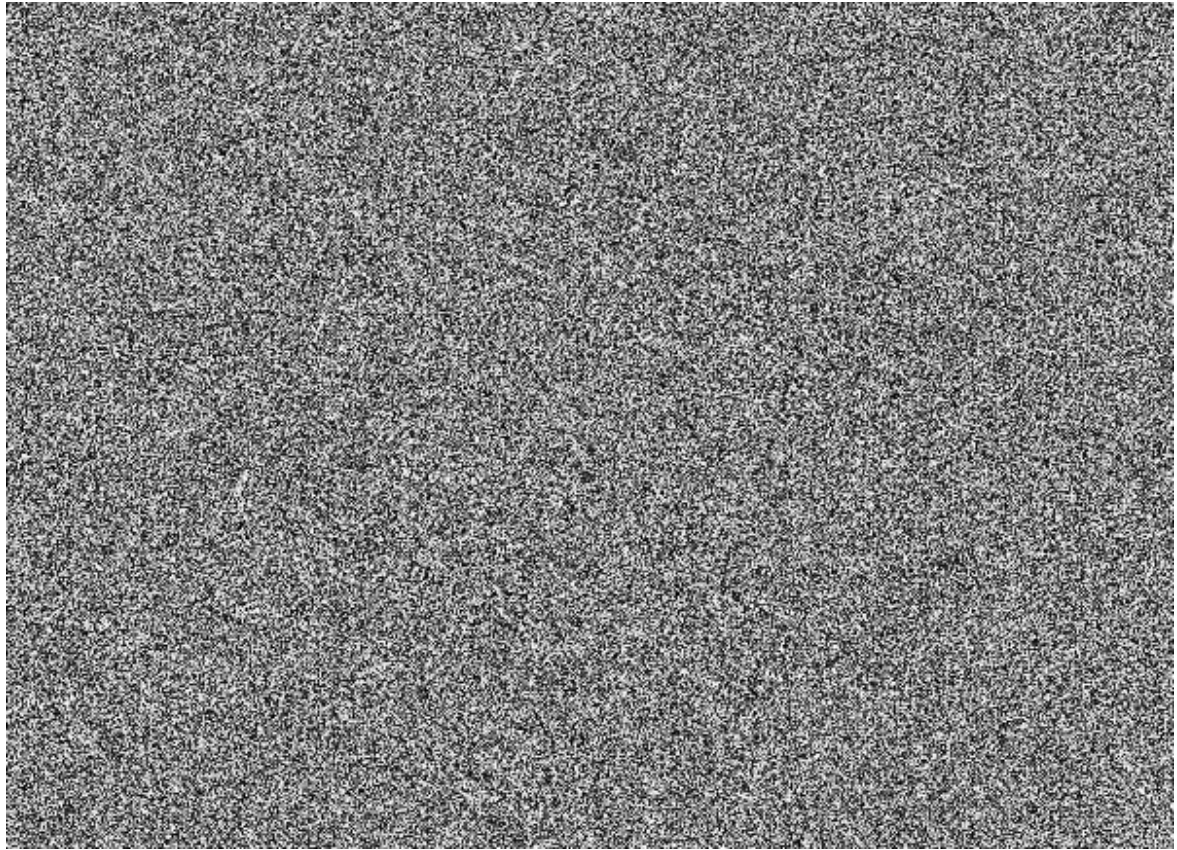


Figure 2-5 CBC Picture Encryption

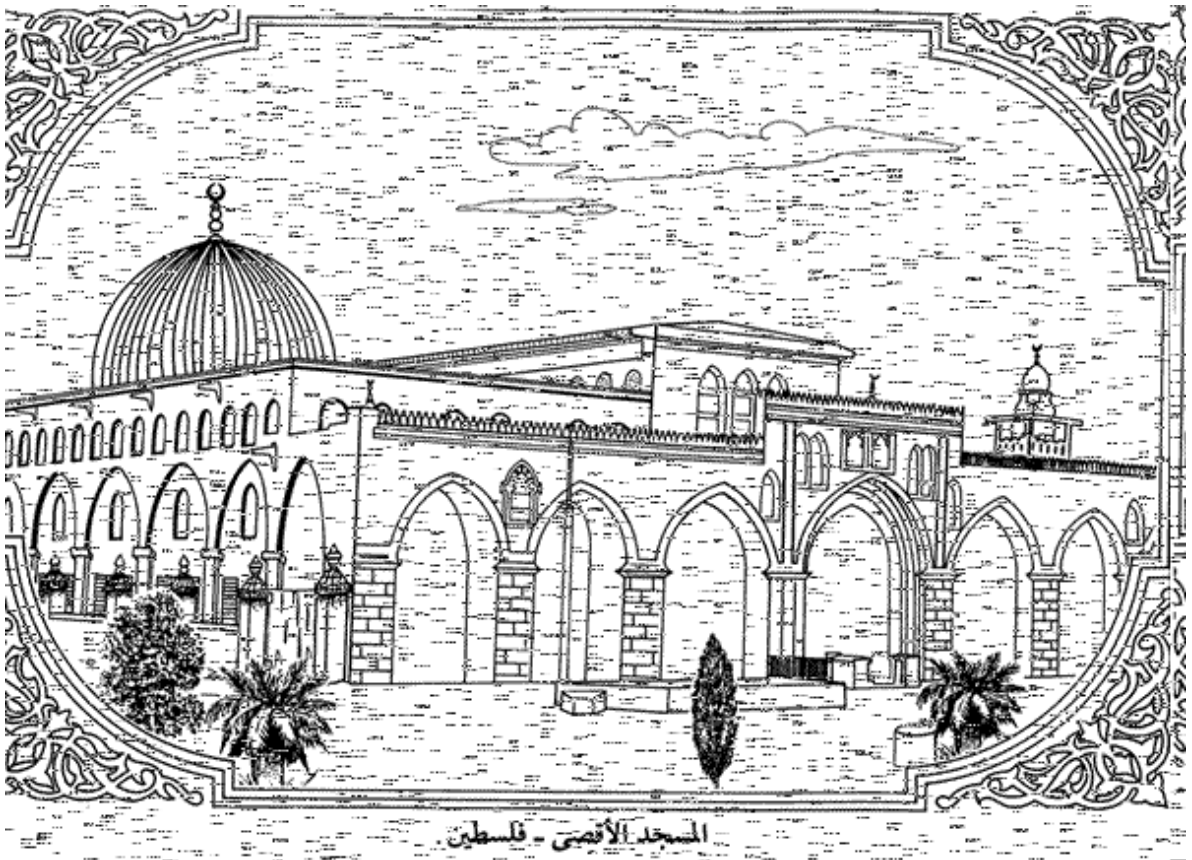


Figure 2-6 CBC Picture Decryption