

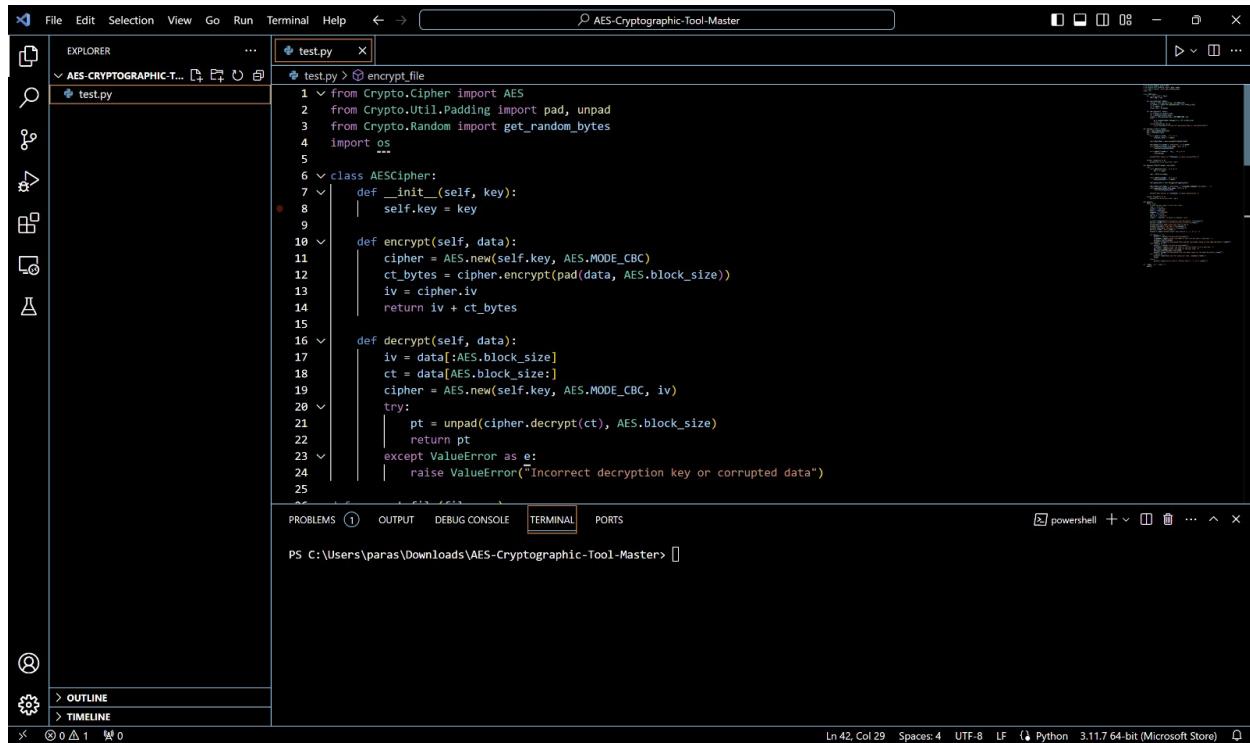
Mandar Angchekar 386916341
Paras Suri 5643363582

Intro to Cryptography

User Manual

Step 1

Open the project folder in a code editor (VS code, Sublime, Atom, etc)
Open terminal and navigate to the project directory.



The screenshot shows a Microsoft Visual Studio Code interface. The title bar says "AES-Cryptographic-Tool-Master". The left sidebar has icons for Explorer, Search, and others. The main area shows a Python file named "test.py" with the following code:

```
1  from Crypto.Cipher import AES
2  from Crypto.Util.Padding import pad, unpad
3  from Crypto.Random import get_random_bytes
4  import os
5
6  class AESCipher:
7      def __init__(self, key):
8          self.key = key
9
10     def encrypt(self, data):
11         cipher = AES.new(self.key, AES.MODE_CBC)
12         ct_bytes = cipher.encrypt(pad(data, AES.block_size))
13         iv = cipher.iv
14         return iv + ct_bytes
15
16     def decrypt(self, data):
17         iv = data[:AES.block_size]
18         ct = data[AES.block_size:]
19         cipher = AES.new(self.key, AES.MODE_CBC, iv)
20         try:
21             pt = unpad(cipher.decrypt(ct), AES.block_size)
22             return pt
23         except ValueError as e:
24             raise ValueError("Incorrect decryption key or corrupted data")
25
```

The bottom status bar shows "PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master>".

Before running the code, install the required package - Crypto
Run the following command:
pip install pycryptodome

Mandar Angchekar 386916341

Paras Suri 5643363582

The screenshot shows the VS Code interface. The Explorer sidebar on the left has a folder named 'AES-CRYPTOGRAPHIC-T...' expanded, showing files 'test.py' and 'test.txt'. The 'test.py' file is open in the main editor area, displaying Python code for AES encryption and decryption. Below the editor is a terminal window showing the command 'pip install pycryptodome' being run, with output indicating the package is already installed. The status bar at the bottom shows the file is in 'Plain Text' mode.

```
1 from Crypto.Cipher import AES
2 from Crypto.Util.Padding import pad, unpad
3 from Crypto.Random import get_random_bytes
4 import os
5
6 class AESCipher:
7     def __init__(self, key):
8         self.key = key
9
10    def encrypt(self, data):
11        cipher = AES.new(self.key, AES.MODE_CBC)
12        ct_bytes = cipher.encrypt(pad(data, AES.block_size))
13        iv = cipher.iv
14        return iv + ct_bytes
15
16    def decrypt(self, data):
17        iv = data[:AES.block_size]
18        ct = data[AES.block_size:]
19        cipher = AES.new(self.key, AES.MODE_CBC, iv)
20        try:
21            pt = unpad(cipher.decrypt(ct), AES.block_size)
22        except ValueError as e:
23            raise ValueError("Incorrect decryption key or corrupted data")
24
25
```

PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master> pip install pycryptodome
Requirement already satisfied: pycryptodome in c:\users\paras\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (3.19.0)
[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: C:\Users\paras\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip
PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master>

This command will install **pycryptodome** and its dependencies, allowing you to use the **Crypto.Cipher**, **Crypto.Util.Padding**, and **Crypto.Random** modules in your Python code.

Make a file in the same directory that you want to encrypt. The file can be of any type eg .txt, .csv, .png etc.

The screenshot shows the VS Code interface. The Explorer sidebar on the left has a folder named 'AES-CRYPTOGRAPHIC-TOOL-MASTER' expanded, showing files 'test.py' and 'test.txt'. The 'test.txt' file is open in the main editor area, containing the text 'This is Cryptography project.' Below the editor is a terminal window showing the command 'PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master>'.

1 This is Cryptography project.

Step 2:

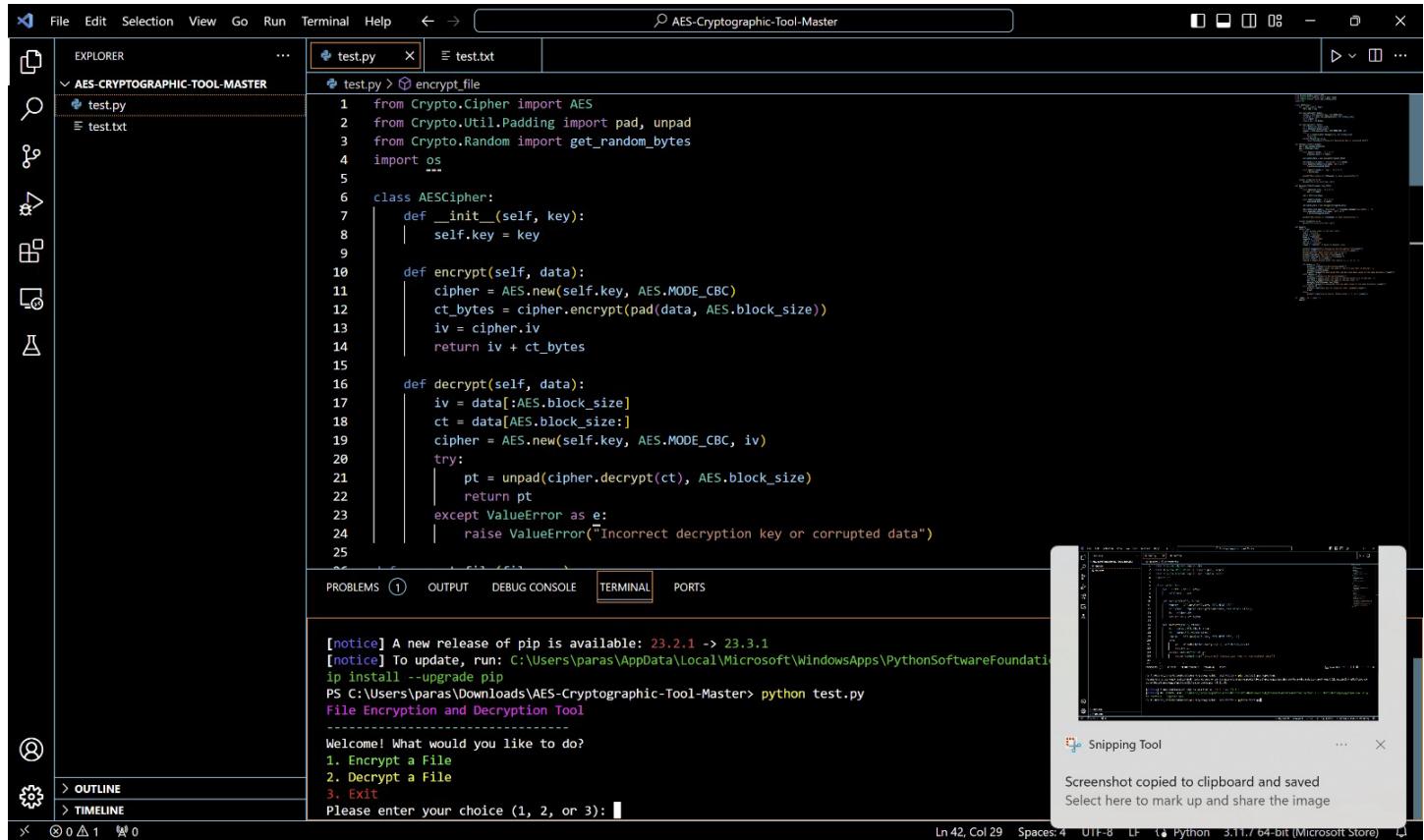
Run the python code file named "test.py".

Mandar Angchekar 386916341

Paras Suri 5643363582

Run the following command:

python test.py



AES-Cryptographic-Tool-Master

File Edit Selection View Go Run Terminal Help

EXPLORER

AES-CRYPTOGRAPHIC-TOOL-MASTER

test.py test.txt

```
1 from Crypto.Cipher import AES
2 from Crypto.Util.Padding import pad, unpad
3 from Crypto.Random import get_random_bytes
4 import os
5
6 class AESCipher:
7     def __init__(self, key):
8         self.key = key
9
10    def encrypt(self, data):
11        cipher = AES.new(self.key, AES.MODE_CBC)
12        ct_bytes = cipher.encrypt(pad(data, AES.block_size))
13        iv = cipher.iv
14        return iv + ct_bytes
15
16    def decrypt(self, data):
17        iv = data[:AES.block_size]
18        ct = data[AES.block_size:]
19        cipher = AES.new(self.key, AES.MODE_CBC, iv)
20        try:
21            pt = unpad(cipher.decrypt(ct), AES.block_size)
22        except ValueError as e:
23            raise ValueError("Incorrect decryption key or corrupted data")
24
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[notice] A new release of pip is available: 23.2.1 > 23.3.1
[notice] To update, run: C:\Users\paras\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation3.11\pip install --upgrade pip
PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master> python test.py
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3):
```

Snipping Tool

Screenshot copied to clipboard and saved

Select here to mark up and share the image

Ln 42, Col 29 Spaces:4 UTF-8 LF Python 3.11 / 64-bit (Microsoft Store)

Step 3:

To encrypt a file type 1 and press Enter

Mandar Angchekar 386916341

Paras Suri 5643363582

AES-CRYPTOGRAPHIC-TOOL-MASTER

File Edit Selection View Go Run Terminal Help

EXPLORER

AES-CRYPTOGRAPHIC-TOOL-MASTER

test.py test.txt

```
1 from Crypto.Cipher import AES
2 from Crypto.Util.Padding import pad, unpad
3 from Crypto.Random import get_random_bytes
4 import os
5
6 class AESCipher:
7     def __init__(self, key):
8         self.key = key
9
10    def encrypt(self, data):
11        cipher = AES.new(self.key, AES.MODE_CBC)
12        ct_bytes = cipher.encrypt(pad(data, AES.block_size))
13        iv = cipher.iv
14        return iv + ct_bytes
15
16    def decrypt(self, data):
17        iv = data[:AES.block_size]
18        ct = data[AES.block_size:]
19        cipher = AES.new(self.key, AES.MODE_CBC, iv)
20        try:
21            pt = unpad(cipher.decrypt(ct), AES.block_size)
22        return pt
23    except ValueError as e:
24        raise ValueError("Incorrect decryption key or corrupted data")
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ip install --upgrade pip
PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master> python test.py
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3): 1
File Encryption
Enter the name of the file you want to encrypt: 
```

Snipping Tool

Screenshot copied to clipboard and saved
Select here to mark up and share the image

Ln 42, Col 29 Spaces: 4 UTF-8 LF Python 3.11.7 64-bit (Microsoft Store)

Enter the file you want to encrypt. The program can take any type of files as input.

Please type the file name along with the extension (.txt, .csv, .png etc.) that you created in step 1.

Press Enter.

Mandar Angchekar 386916341

Paras Suri 5643363582

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar reads "AES-Cryptographic-Tool-Master". The left sidebar has icons for Explorer, Search, Find, and Outline/Timeline. The main area shows a file tree under "AES-CRYPTOGRAPHIC-TOOL-MASTER" with files "test.py", "encrypted_test.txt", "test.txt", and "test.txt.key". The "test.py" tab is active, displaying the following Python code:

```
1  from Crypto.Cipher import AES
2  from Crypto.Util.Padding import pad, unpad
3  from Crypto.Random import get_random_bytes
4  import os
5
6  class AESCipher:
7      def __init__(self, key):
8          self.key = key
9
10     def encrypt(self, data):
11         cipher = AES.new(self.key, AES.MODE_CBC)
12         ct_bytes = cipher.encrypt(pad(data, AES.block_size))
13         iv = cipher.iv
14         return iv + ct_bytes
15
16     def decrypt(self, data):
17         iv = data[:AES.block_size]
18         ct = data[AES.block_size:]
19         cipher = AES.new(self.key, AES.MODE_CBC, iv)
20         try:
21             pt = unpad(cipher.decrypt(ct), AES.block_size)
22         except ValueError as e:
23             raise ValueError("Incorrect decryption key or corrupted data")
24
25
```

The bottom status bar shows "Ln 42, Col 29" and "python3.11". The terminal pane below the code editor displays the output of the script:

```
File Encryption
Enter the name of the file you want to encrypt: test.txt
Encryption of test.txt is done successfully.
The encrypted file and key have been saved in the same directory.
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3):
```

This should create two files in the same directory namely:

- 1) Encrypted file: In this example encrypted_test.txt

Mandar Angchekar 386916341

Paras Suri 5643363582

A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "AES-Cryptographic-Tool-Master". The left sidebar shows an "EXPLORER" view with files: "test.py", "encrypted_test.txt", "test.txt", and "test.txt.key". The "TERMINAL" tab is selected, displaying the following terminal output:

```
File Encryption
Enter the name of the file you want to encrypt: test.txt
Encryption of test.txt is done successfully.
The encrypted file and key have been saved in the same directory.
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3):
```

The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4" and "UTF-8" and "CRLF" and "Plain Text".

2) Key: In this example test.txt.key

A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "AES-Cryptographic-Tool-Master". The left sidebar shows an "EXPLORER" view with files: "test.py", "test.txt.key", "test.txt", and "test.txt". The "TERMINAL" tab is selected, displaying the following terminal output:

```
File Encryption
Enter the name of the file you want to encrypt: test.txt
Encryption of test.txt is done successfully.
The encrypted file and key have been saved in the same directory.
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3):
```

A floating window titled "Snipping Tool" is visible in the bottom right corner, showing a screenshot of the terminal window.

The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4" and "UTF-8" and "CRLF" and "Plain Text".

Mandar Angchekar 386916341
Paras Suri 5643363582

Step 4:

To decrypt a file type 2 and press Enter

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files: test.py, test.txt.key, and test.txt.
- Terminal:** Displays Python code for AES encryption and decryption using the Crypto library. The code defines a class AESCipher with methods for encrypting and decrypting data using CBC mode.
- Output:** Shows the terminal output:

```
Encryption of test.txt is done successfully.  
The encrypted file and key have been saved in the same directory.  
File Encryption and Decryption Tool  
-----  
Welcome! What would you like to do?  
1. Encrypt a File  
2. Decrypt a File  
3. Exit  
Please enter your choice (1, 2, or 3): 2
```
- Terminal:** Shows the user entering "2" to select the decryption option.
- Output:** Shows the terminal output:

```
File Decryption  
Enter the name of the encrypted file to decrypt: test.txt
```
- Status Bar:** Shows the current file is python3.11, line 42, column 29, and other system information.

Enter the name of the encrypted file that was created in the directory in step3. In this example
encrypted_test.txt
Press Enter

Mandar Angchekar 386916341

Paras Suri 5643363582

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "AES-CRYPTOGRAPHIC-TOOL-MASTER" containing files "test.py", "encrypted.test.txt", "test.txt", and "test.txtkey".
- Terminal:** The terminal tab is active, displaying the output of a Python script.

```
The encrypted file and key have been saved in the same directory.
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3): 2

File Decryption
Enter the name of the encrypted file to decrypt: encrypted_test.txt
Enter the name of the key file: 
```
- Status Bar:** Shows "Ln 42, Col 29" and "Python 3.11.7 64-bit (Microsoft Store)".

Enter the name of the fey file that was created in the directory in step 3. In this example test.text.key
Press Enter

Mandar Angchekar 386916341

Paras Suri 5643363582

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "AES-CRYPTOGRAPHIC-TOOL-MASTER" folder: "decrypted_test.txt", "encrypted_test.txt", "test.py", "test.txt", and "test.txt.key".
- Terminal:** The current tab is "test.py". The code implements AES encryption and decryption using CBC mode. It reads a key from "test.txt.key", encrypts "test.txt" to "encrypted_test.txt", and decrypts it back to "decrypted_test.txt".

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from Crypto.Random import get_random_bytes
import os

class AESCipher:
    def __init__(self, key):
        self.key = key

    def encrypt(self, data):
        cipher = AES.new(self.key, AES.MODE_CBC)
        ct_bytes = cipher.encrypt(pad(data, AES.block_size))
        iv = cipher.iv
        return iv + ct_bytes

    def decrypt(self, data):
        iv = data[:AES.block_size]
        ct = data[AES.block_size:]
        cipher = AES.new(self.key, AES.MODE_CBC, iv)
        try:
            pt = unpad(cipher.decrypt(ct), AES.block_size)
            return pt
        except ValueError as e:
            raise ValueError("Incorrect decryption key or corrupted data")
```
- Terminal Output:**

```
File Decryption
Enter the name of the encrypted file to decrypt: encrypted_test.txt
Enter the name of the key file: test.txt.key
Decryption of encrypted_test.txt is done successfully.
The decrypted file has been saved in the same directory.
File Encryption and Decryption Tool
-----
Welcome! What would you like to do?
1. Encrypt a File
2. Decrypt a File
3. Exit
Please enter your choice (1, 2, or 3):
```
- Status Bar:** Shows "Ln 42, Col 29" and "Python 3.11.7 64-bit (Microsoft Store)".

This should create the decrypted file in the same directory:

- 1) Decrypted file: In this example decrypted_test.txt

Note: The decrypted file created using the encrypted file and the key should be the same as the original file you created in the step 1.

Enter 3 to exit

Mandar Angchekar 386916341

Paras Suri 5643363582

The screenshot shows a terminal window titled "AES-Cryptographic-Tool-Master" running in a code editor interface. The terminal output is as follows:

```
1 This is Cryptography project.  
Enter the name of the key file: test.txt.key  
Decryption of encrypted_test.txt is done successfully.  
The decrypted file has been saved in the same directory.  
File Encryption and Decryption Tool  
-----  
Welcome! What would you like to do?  
1. Encrypt a File  
2. Decrypt a File  
3. Exit  
Please enter your choice (1, 2, or 3): 3  
Thank you for using our tool. Goodbye!  
PS C:\Users\paras\Downloads\AES-Cryptographic-Tool-Master>
```

A tooltip from the Snipping Tool is visible in the bottom right corner, stating: "Screenshot copied to clipboard and saved Select here to mark up and share the image".