**ADVANCED ENCRYPTION STANDARD (AES):**

**CODE:**

from Crypto.Cipher import AES

from Crypto.Random import get\_random\_bytes

import base64

def pad(data):

return data + (16 - len(data) % 16) \* chr(16 - len(data) % 16)

def unpad(data):

return data[:-data[-1]] # Fixed unpadding function

def encrypt\_AES(plain\_text, key):

cipher = AES.new(key, AES.MODE\_CBC)

iv = cipher.iv

encrypted\_text = cipher.encrypt(pad(plain\_text).encode())

return base64.b64encode(iv + encrypted\_text).decode()

def decrypt\_AES(encrypted\_text, key):

encrypted\_data = base64.b64decode(encrypted\_text)

iv = encrypted\_data[:16]

cipher = AES.new(key, AES.MODE\_CBC, iv)

decrypted\_text = unpad(cipher.decrypt(encrypted\_data[16:])).decode()

return decrypted\_text

# Example Usage

key = get\_random\_bytes(16) # 128-bit key

plain\_text = "Network Security Assignment"

encrypted\_text = encrypt\_AES(plain\_text, key)

decrypted\_text = decrypt\_AES(encrypted\_text, key)

print(f"Original Text: {plain\_text}")

print(f"Encrypted Text: {encrypted\_text}")

print(f"Decrypted Text: {decrypted\_text}")

**OUTPUT:**

Original Text: Network Security Assignment

EncryptedText: DhD1GSdCA6bjQZJKHc1j3DcXFR/GRILrcI6hIqm0wR2e9qLM75SotYrMNbMf8/Ff

Decrypted Text: Network Security Assignment

**EXPLANATION OF THE IMPLEMENTATION:**

#### **1. Importing Required Libraries**

* Crypto.Cipher.AES: Provides AES encryption and decryption.
* Crypto.Random.get\_random\_bytes: Generates a secure random key.
* base64: Used to encode and decode the encrypted text for easier storage.

#### **2. Padding and Unpadding Functions**

* AES operates on fixed-size blocks (16 bytes). If the input text is shorter than 16 bytes, we pad it.
* This function appends extra characters to make the length a multiple of 16.
* This removes the extra padding when decrypting.

#### **3. Encryption Function**

* AES.new(key, AES.MODE\_CBC): Creates a new AES cipher in CBC mode.
* cipher.iv: Generates an Initialization Vector (IV) to ensure encryption uniqueness.
* cipher.encrypt(): Encrypts the padded text.
* base64.b64encode(): Encodes the IV and encrypted text to make it easier to store or transmit.

#### **4. Decryption Function**

* Decodes the Base64-encoded encrypted text.
* Extracts the IV from the first 16 bytes.
* Uses the same key and IV to decrypt the text.
* Removes padding and returns the original text.

#### **5. Example Usage**

* Generates a random 128-bit AES key.
* Encrypts the given plaintext.
* Decrypts the encrypted text.
* Prints the results.

**Output:**

The program displays the ciphertext in hexadecimal format. It then decrypts the ciphertext and prints the original plaintext.