**Implementation of Symmetric cipher algorithm (AES)**

**R.Shiny**

**AP25122050012**

**CODE:**

from Crypto.Cipher import AES  
from Crypto.Random import get\_random\_bytes  
from Crypto.Util.Padding import pad, unpad  
  
# AES setup  
key\_aes = get\_random\_bytes(16)  # AES-128  
cipher\_aes = AES.new(key\_aes, AES.MODE\_CBC)  
plaintext\_aes = b"HelloAESWorld"  
ciphertext\_aes = cipher\_aes.encrypt(pad(plaintext\_aes, AES.block\_size))  
  
# Decryption  
decipher\_aes = AES.new(key\_aes, AES.MODE\_CBC, cipher\_aes.iv)  
decrypted\_aes = unpad(decipher\_aes.decrypt(ciphertext\_aes), AES.block\_size)  
  
print("Ciphertext:", ciphertext\_aes)  
print("Decrypted:", decrypted\_aes)

**OUTPUT:**

****

**Implementation of Symmetric cipher algorithm (RC4)**

**CODE :**

**def rc4(key, data):**

**S = list(range(256))**

**j = 0**

**out = []**

**# Key Scheduling Algorithm (KSA)**

**for i in range(256):**

**j = (j + S[i] + key[i % len(key)]) % 256**

**S[i], S[j] = S[j], S[i]**

**# Pseudo-Random Generation Algorithm (PRGA)**

**i = j = 0**

**for char in data:**

**i = (i + 1) % 256**

**j = (j + S[i]) % 256**

**S[i], S[j] = S[j], S[i]**

**K = S[(S[i] + S[j]) % 256]**

**out.append(char ^ K)**

**return bytes(out)**

**# Original message**

**plaintext = "Hello Shiny, RC4 is fast!"**

**data = plaintext.encode('utf-8')**

**key = b'secretkey'**

**# Encrypt**

**ciphertext = rc4(key, data)**

**print("RC4 Encrypted (hex):", ciphertext.hex())**

**# Decrypt (same function)**

**decrypted = rc4(key, ciphertext)**

**print("RC4 Decrypted:", decrypted.decode('utf-8'))**

**Output :**

****