```
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import padding
import os
import binascii
```

```
# Function to encrypt data
def encrypt_data(data, key):
  iv = os.urandom(16) # Generate random IV
  cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
  encryptor = cipher.encryptor()

  # Pad the data to be a multiple of 128 bits (16 bytes)
  padder = padding.PKCS7(128).padder()
  padded_data = padder.update(data.encode()) + padder.finalize()

  # Encrypt the padded data
  ciphertext = encryptor.update(padded_data) + encryptor.finalize()

  # Return IV and ciphertext as hex strings
  return binascii.hexlify(iv).decode('utf-8'), binascii.hexlify(ciphertext).decode(
```

```
# Function to decrypt data
def decrypt_data(iv, ciphertext, key):
  iv = binascii.unhexlify(iv) # Convert the hex-encoded IV back to bytes

  ciphertext = binascii.unhexlify(ciphertext) # Convert the hex-encoded ciphertext

  # Create the cipher object for decryption
  cipher=Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
  decryptor = cipher.decryptor()

  # Decrypt the ciphertext
  decrypted_data = decryptor.update(ciphertext) + decryptor.finalize()

  # Unpad the decrypted data
  unpadder = padding.PKCS7(128).unpadder()   # Create the unpadder
  unpadded_data = unpadder.update(decrypted_data) + unpadder.finalize()   # Remove

  # Return the decrypted string
  return unpadded_data.decode()


# Main program
if __name__=="__main__":
  key = os.urandom(32)  # 256-bit key for AES
  data = "This is a Secret Message"
  print("Original Data:", data)

  iv, ciphertext = encrypt_data(data, key)
  print("Encrypted Data (IV):", iv)
```

```
    print("Encrypted Data (Ciphertext):", ciphertext)

    decrypted_data = decrypt_data(iv, ciphertext, key)
    print("Decrypted Data:", decrypted_data)
```

```
Original Data: This is a Secret Message
Encrypted Data (IV): 1ec138b3f1e94cdb0871b6d02ccb9e1b
Encrypted Data (Ciphertext): e61d6e286650e81a0718e8c4707d1b85e402e3df3ca45ecefebd37f6
Decrypted Data: This is a Secret Message
```

Start coding or generate with AI.