

Cifrado simétrico AES (Advanced Encryption Standard) en Python

El algoritmo AES es ampliamente utilizado para cifrado simétrico debido a su alta seguridad y eficiencia. En este ejemplo, utilizaremos el modo de operación CBC (Cipher Block Chaining) para cifrar y descifrar un mensaje.

Este ejemplo generará una clave AES, cifrará un mensaje y luego lo descifrá nuevamente para obtener el mensaje original. El resultado mostrará el mensaje original, el mensaje cifrado en formato de bytes y el mensaje descifrado que debería ser igual al original.

[crayon-6a9d4eb1340fb476923416/]

```
Añadir celda de texto
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.primitives import padding
from cryptography.hazmat.backends import default_backend
import os

def generate_aes_key():
    return os.urandom(16) # Tamaño de clave: 128 bits (AES-128)

def encrypt_aes(text, key):
    iv = os.urandom(16) # Genera un nuevo IV
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
    encryptor = cipher.encryptor()
    padder = padding.PKCS7(algorithms.AES.block_size).padder()
    padded_data = padder.update(text.encode('utf-8')) + padder.finalize()
    encrypted_text = encryptor.update(padded_data) + encryptor.finalize()
    return iv + encrypted_text

def decrypt_aes(encrypted_data, key):
    iv = encrypted_data[:16] # Tamaño del vector de inicialización (IV)
    encrypted_text = encrypted_data[16:]
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
    decryptor = cipher.decryptor()
    unpadder = padding.PKCS7(algorithms.AES.block_size).unpadder()
    padded_data = decryptor.update(encrypted_text) + decryptor.finalize()
    decrypted_text = unpadder.update(padded_data) + unpadder.finalize()
    return decrypted_text.decode('utf-8')

def main():
    original_message = "¡Hola, este es un ejemplo de cifrado AES en Python!"
    aes_key = generate_aes_key()

    encrypted_data = encrypt_aes(original_message, aes_key)
    decrypted_message = decrypt_aes(encrypted_data, aes_key)

    print(f"Mensaje original: {original_message}")
    print(f"Mensaje cifrado: {encrypted_data.hex()}")
    print(f"Mensaje descifrado: {decrypted_message}")

if __name__ == "__main__":
    main()
```

Mensaje original: ¡Hola, este es un ejemplo de cifrado AES en Python!
Mensaje cifrado: 79b38293d78a5fb210a99da70d23660caceb0f50b793816d5833ac630dcfbcef27319bfed1b01ce0e4d62eeecb8cd50cfc0ff2d9725537f8d3a53c2ff945866cf48cb1d0465b8300a317afe698d64
Mensaje descifrado: ¡Hola, este es un ejemplo de cifrado AES en Python!