Informe Laboratorio 4

Sección 1

Bruno Figueroa
e-mail: bruno.figueroa@mail.udp.cl

Noviembre de 2025

Índice

1. Descripción de actividades **2**

2. Desarrollo de actividades según criterio de rúbrica **3**
2.1. Investiga y documenta los tamaños de clave e IV 3
2.2. Solicita datos de entrada desde la terminal 4
2.3. Valida y ajusta la clave según el algoritmo 5
2.4. Implementa el cifrado y descifrado en modo CBC 5
2.5. Compara los resultados con un servicio de cifrado online 7
2.6. Describe la aplicabilidad del cifrado simétrico en la vida real 8

3. Referencias **9**

1. Descripción de actividades

Desarrollar un programa en Python utilizando la librería pycrypto para cifrar y descifrar mensajes con los algoritmos DES, AES-256 y 3DES, permitiendo la entrada de la key, vector de inicialización y el texto a cifrar desde la terminal.

Instrucciones:

1. Investigación
   - Investigue y documente el tamaño en bytes de la clave y el vector de inicialización (IV) requeridos para los algoritmos DES, AES-256 y 3DES. Mencione las principales diferencias entre cada algoritmo, sea breve.
2. El programa debe solicitar al usuario los siguientes datos desde la terminal
   - Key correspondiente a cada algoritmo.
   - Vector de Inicialización (IV) para cada algoritmo.
   - Texto a cifrar.
3. Validación y ajuste de la clave
   - Si la clave ingresada es menor que el tamaño necesario para el algoritmo complete los bytes faltantes agregando bytes adicionales generados de manera aleatoria (utiliza get_random_bytes).
   - Si la clave ingresada es mayor que el tamaño requerido, trunque la clave a la longitud necesaria.
   - Imprima la clave final utilizada para cada algoritmo después de los ajustes.
4. Cifrado y Descifrado
   - Implemente una función para cada algoritmo de cifrado y descifrado (DES, AES-256, y 3DES). Use el modo CBC para todos los algoritmos.
   - Asegúrese de utilizar el IV proporcionado por el usuario para el proceso de cifrado y descifrado.
   - Imprima tanto el texto cifrado como el texto descifrado.
5. Comparación con un servicio de cifrado online
   - Selecciona uno de los tres algoritmos (DES, AES-256 o 3DES), ingrese el mismo texto, key y vector de inicialización en una página web de cifrado online.
   - Compare los resultados de tu programa con los del servicio online. Valide si el resultado es el mismo y fundamente su respuesta.
6. Aplicabilidad en la vida real

- Describa un caso, situación o problema donde usaría cifrado simétrico. Defina que algoritmo de cifrado simétrico recomendaría justificando su respuesta.
- Suponga que la recomendación que usted entrego no fue bien percibida por su contraparte y le pide implementar hashes en vez de cifrado simétrico. Argumente cuál sería su respuesta frente a dicha solicitud.

2. Desarrollo de actividades según criterio de rúbrica

2.1. Investiga y documenta los tamaños de clave e IV

Para cada algoritmo se investigó el tamaño de clave y vector de inicialización (IV) requerido según Alanazi et al. (2010) [**alanazi2010comparative**]:

- **DES**: utiliza una clave de 8 bytes (64 bits), de los cuales 56 bits son efectivos y 8 bits se reservan para verificación de paridad. El IV es de 8 bytes, ya que DES opera en bloques de 64 bits. La longitud de la clave era suficiente para la seguridad de los años 70, pero hoy es vulnerable a ataques de fuerza bruta.
- **3DES**: combina tres aplicaciones de DES con tres claves de 56 bits (168 bits efectivos) o dos claves de 56 bits (112 bits efectivos). El IV también es de 8 bytes, consistente con el tamaño de bloque de 64 bits. Esto incrementa la seguridad frente a ataques que DES no puede resistir.
- **AES-256**: utiliza una clave de 32 bytes (256 bits) y bloques de 16 bytes (128 bits), por lo que el IV debe ser de 16 bytes en modo CBC. La longitud de la clave y el tamaño de bloque más grandes permiten una mayor seguridad y resistencia frente a ataques de fuerza bruta y criptanalíticos.

Factors	**AES-256**	**3DES**	**DES**
Tamaño de key	256 bits	168 o 192 bits	56 o 64 bits
Tamaño de IV	128 bits	64 bits	64 bits

Tabla 1: Comparación de tamaño de clave y bloque (IV) para DES, 3DES y AES-256. Los valores son los utilizados en el código desarrollado.

En resumen, la principal diferencia entre los algoritmos radica en la longitud de la clave y el tamaño del bloque. DES, con clave corta y bloque pequeño, era suficiente para la seguridad de los años 70, pero hoy es vulnerable a ataques de fuerza bruta; 3DES mejora la seguridad mediante la triple aplicación de DES; y AES-256 ofrece un estándar moderno con clave y bloque más largos, garantizando resistencia a ataques contemporáneos.

2.2. Solicita datos de entrada desde la terminal

Para comenzar, se tiene que mencionar que el código fue generado con la ayuda de ChatGPT, luego fue depurado manualmente para asegurar su correcto funcionamiento. A continuación, se muestra la imagen del prompt utilizado inicialmente antes de iniciar el proceso de debugging.

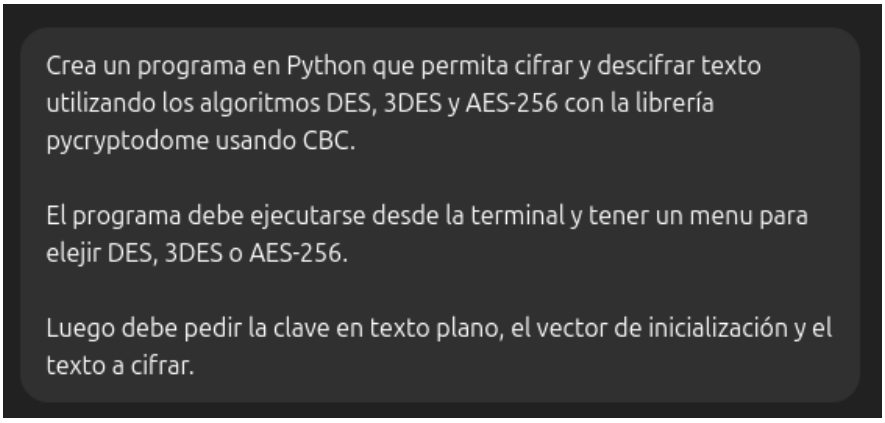

Figura 1: Prompt inicial utilizado en ChatGPT para generar la primera versión del código.

Ahora, ya explicado el uso inicial con Chat GPT, se explica el funcionamiento de la aplicacion.

Inicialmente, el programa solicita de manera interactiva los siguientes parámetros, utilizando un menú de selección inicial, para luego pedir los strings necesarios uno a uno:

- El tipo de algoritmo (DES, 3DES o AES-256).
- La clave (key).
- El vector de inicialización (IV).
- El texto a cifrar.

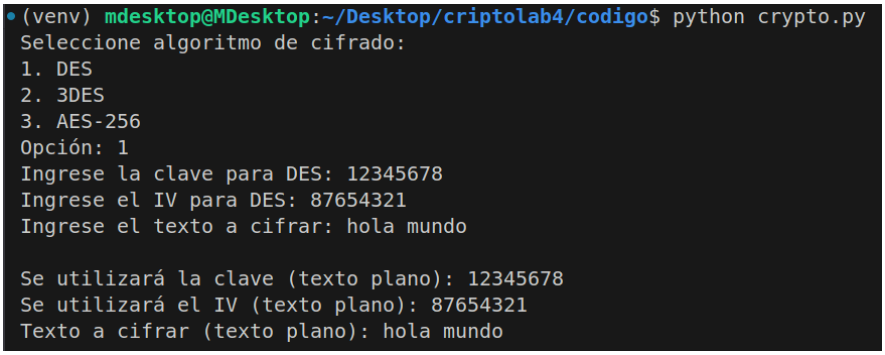

Figura 2: Imagen del programa solicitando los datos desde la terminal.

La figura 2 muestra cómo se realiza la selección del algoritmo y la petición de parámetros por parte del usuario, para su posterior uso en el proceso de cifrado y descifrado.

2.3. Valida y ajusta la clave según el algoritmo

La función `ensure_length` verifica si la clave o el IV cumplen el tamaño necesario. Si no lo hacen, el programa:

- Completa con bytes aleatorios cuando es menor, utilizando la funcion Crypto.Random.
- Trunca cuando es mayor.

De esta forma, cada algoritmo recibe valores válidos sin errores.

```
Ingrese la clave para DES: 123
Ingrese el IV para DES: 12345678901234567890
Ingrese el texto a cifrar: asereje aja eje

Se utilizará la clave (texto plano): 123M3\p>
Se utilizará el IV (texto plano, truncado): 12345678
Texto a cifrar (texto plano): asereje aja eje
```

Figura 3: Ajuste automático de clave e IV según el tamaño requerido por el algoritmo.

En la figura 3 se ilustra este proceso: se ingresa una clave menor a la requerida (“123”), que se rellena automáticamente con padding aleatorio (“`123M3\p>`”). De manera similar, se entrega un IV más largo de lo necesario (“12345678901234567890”), que el programa trunca a la longitud correcta (“12345678”). Esto asegura que la clave y el IV cumplan con los tamaños esperados por el algoritmo seleccionado (DES).

2.4. Implementa el cifrado y descifrado en modo CBC

El programa cifra y descifra utilizando el modo **CBC (Cipher Block Chaining)**. El texto cifrado se muestra en formato **Base64** para hacerlo legible, y el descifrado retorna el texto original, demostrando que el proceso es reversible.

En la figura 4 se muestra la función utilizada para cifrar y descifrar los algoritmos simétricos en modo CBC. Al crear el objeto del cifrado se especifica el modo mediante `.MODE_CBC` (destacado en color naranja), asegurando que se aplique correctamente la operación de encadenamiento de bloques. Esta función permite recibir cualquier algoritmo, la clave, el IV y el texto plano, cifrando y descifrando automáticamente, y retorna tanto el texto cifrado como el descifrado.

```python
def encrypt_decrypt(cipher_cls, key, iv, plaintext, block_size):
    cipher = cipher_cls.new(key, cipher_cls.MODE_CBC, iv)
    ct = cipher.encrypt(pad(plaintext.encode(), block_size))
    decipher = cipher_cls.new(key, cipher_cls.MODE_CBC, iv)
    pt = unpad(decipher.decrypt(ct), block_size)
    return ct, pt
```

Figura 4: Snippet de cifrado y descifrado en modo CBC, destacando el uso de `.MODE_CBC`.

A continuación se muestran los resultados del cifrado y descifrado de cada algoritmo utilizando esta función.

```
(venv) mdesktop@MDesktop:~/Desktop/criptolab4/codigo$ python crypto.py
Seleccione algoritmo de cifrado:
1. DES
2. 3DES
3. AES-256
Opción: 1
Ingrese la clave para DES: 12345678
Ingrese el IV para DES: 87654321
Ingrese el texto a cifrar: hola mundo

Se utilizará la clave (texto plano): 12345678
Se utilizará el IV (texto plano): 87654321
Texto a cifrar (texto plano): hola mundo

Texto cifrado (DES, Base64): kDkYRy1qIJrfYu9g3AP3HQ==
Texto descifrado (DES): hola mundo
```

Figura 5: Resultado del proceso de cifrado y descifrado con DES.

Como se observa en la figura 5, se utiliza el algoritmo DES para encriptar el texto "hola mundo" con la clave "12345678" y el IV "87654321". El mensaje cifrado resultante es "kDkYRy1qIJrfYu9g3AP3HQ==" en Base64. Al aplicar el descifrado, se recupera el texto original "hola mundo", demostrando la reversibilidad del proceso.

```
(venv) mdesktop@MDesktop:~/Desktop/criptolab4/codigo$ python crypto.py
Seleccione algoritmo de cifrado:
1. DES
2. 3DES
3. AES-256
Opción: 2
Ingrese la clave para 3DES: 123456789123456789123456
Ingrese el IV para 3DES: 87654321
Ingrese el texto a cifrar: hola mundo

Se utilizará la clave (texto plano): 123456789123456789123456
Se utilizará el IV (texto plano): 87654321
Texto a cifrar (texto plano): hola mundo

Texto cifrado (3DES, Base64): wBU5zFNS76+A9S+56rXw2g==
Texto descifrado (3DES): hola mundo
```

Figura 6: Resultado del proceso de cifrado y descifrado con 3DES.

La figura 6 muestra cómo 3DES cifra el texto “hola mundo” usando la clave “1234567891234567891234 y un IV de “87654321”. El texto cifrado resultante es “wBU5zFNS76+A9S+56rXw2g==” en Base64. Al descifrarlo, se obtiene nuevamente “hola mundo”, evidenciando que el algoritmo funciona correctamente y mantiene la reversibilidad del mensaje.

```
(venv) mdesktop@MDesktop:~/Desktop/criptolab4/codigo$ python crypto.py
Seleccione algoritmo de cifrado:
1. DES
2. 3DES
3. AES-256
Opción: 3
Ingrese la clave para AES-256: 12345678912345678912345678912345
Ingrese el IV para AES-256: 1234567891234567
Ingrese el texto a cifrar: hola mundo

Se utilizará la clave (texto plano): 12345678912345678912345678912345
Se utilizará el IV (texto plano): 1234567891234567
Texto a cifrar (texto plano): hola mundo

Texto cifrado (AES-256, Base64): frzxXcfY17a5QLU16z7vPA==
Texto descifrado (AES-256): hola mundo
```

Figura 7: Resultado del proceso de cifrado y descifrado con AES-256.

En la figura 7, AES-256 se utiliza para cifrar el texto “hola mundo” con la clave “1234567891234567891 y un IV de “1234567891234567”. El resultado cifrado es “frzxXcfY17a5QLU16z7vPA==” en Base64 y, al descifrarlo, se recupera exactamente el texto original “hola mundo”. Esto confirma que la implementación del modo CBC con AES-256 funciona correctamente y mantiene la integridad del mensaje.

2.5. Compara los resultados con un servicio de cifrado online

Como se puede observar en la figura 8, se probó el algoritmo DES en el sitio web `https://des.codethoi.com/`, usando los mismos parámetros que en la prueba local:

- Texto plano: `hola mundo`
- Clave: `12345678`
- IV: `87654321`
- Modo de encriptado: CBC

El resultado obtenido en Base64 (`k6W0tuul8iOl1qejUbbkUw==`) coincidió exactamente con el generado por el programa local (ver figura 5, DES con “hola mundo”), demostrando que la implementación funciona correctamente y que los resultados son consistentes con un servicio de cifrado online.

Figura 8: Comparación del resultado del programa con el servicio online, mostrando coincidencia exacta.

2.6. Describe la aplicabilidad del cifrado simétrico en la vida real

El cifrado simétrico se aplica cuando ambas partes comparten una misma clave secreta. Por ejemplo, en la transferencia de archivos internos dentro de una empresa. En ese caso se recomienda **AES-256** dado a que cumple con los estandares modernos de seguridad, ya que con DES y 3DES se pueden utilizar ataques de fuerza bruta, o diccionarios para superarlos.

Si la contraparte propone usar funciones hash en lugar de cifrado, se le debe advertir que los hashes no permiten recuperar la información original, ya que son unidireccionales. Y por ende, en casos donde se debe recuperar la informacion inicial, estos no son utiles.

Conclusiones y comentarios

En este laboratorio se implementó un programa en Python para cifrado y descifrado utilizando los algoritmos DES, 3DES y AES-256 en modo CBC. Se verificó que la longitud de la clave y del IV es crítica para la correcta operación de cada algoritmo, y se automatizó el ajuste de estos valores para evitar errores.

Los resultados obtenidos mostraron que los tres algoritmos son capaces de cifrar y descifrar correctamente, manteniendo la integridad del mensaje original, y que los textos cifrados coinciden con los generados por un servicio de cifrado online, demostrando la correcta implementación de estos.

Adicionalmente se puede inferir que DES, por su corta longitud de clave y bloque, es inseguro en la actualidad; 3DES mejora la seguridad mediante la triple aplicación de DES; y AES-256 ofrece un estándar moderno con alta resistencia a ataques de fuerza bruta y criptanalíticos.

Finalmente, se concluye que el cifrado simétrico es útil en escenarios donde la confidencialidad debe mantenerse y ambas partes pueden compartir una clave secreta, siendo AES-256 la opción más recomendable por su seguridad y eficiencia. Además, se reafirma que los hashes no pueden reemplazan al cifrado cuando se necesita recuperar la información original.

3. Referencias

Hamdan O. Alanazi, B. B. Zaidan, A. A. Zaidan, Hamid A. Jalab, M. Shabbir, Y. Al-Nabhani, "New Comparative Study Between DES, 3DES and AES within Nine Factors," *Journal of Computing*, vol. 2, no. 3, 2010. Disponible en: `https://www.semanticscholar.org/paper/New-Comparative-Study-Between-DES%2C-3DES-and-AES-Alanazi-Zaidan/8f5bde83b7`