Cifrado

1. **Lectura de la clave codificada en base64**:

   - El código abre un archivo que contiene la clave codificada en base64 (`clave_archivo`).

   - Decodifica la clave de base64 para obtener la clave real que se utilizará para el cifrado.

2. **Lectura del texto plano del archivo**:

   - Abre otro archivo que contiene el texto plano que se desea cifrar (`texto_archivo`).

   - Lee el contenido del archivo y almacena el texto plano.

3. **Creación del cifrador 3DES en modo CBC**:

   - Se crea un cifrador 3DES en modo Cipher Block Chaining (CBC) con una clave y un vector de inicialización (IV) aleatorio.

   - El IV es necesario para el modo CBC y se genera automáticamente.

NOTA: Modo CBC (Cipher Block Chaining) es una técnica de cifrado que combina cada bloque de texto plano con el resultado del cifrado del bloque anterior.

Funcionamiento:

1. Se genera un vector de inicialización (IV) aleatorio. El IV es un bloque de datos que se utiliza como entrada en la primera ronda de cifrado.
2. El texto plano se divide en bloques de 8 bytes (64 bits).
3. Cada bloque se cifra utilizando la clave 3DES y el IV.
4. El resultado cifrado se combina con el siguiente bloque de texto plano antes de cifrarlo.
5. El proceso se repite hasta que todos los bloques se hayan cifrado.
6. El último bloque cifrado se convierte en el IV para el siguiente bloque.

4. **Relleno del texto plano**:

   - Para que el texto plano sea un múltiplo del tamaño de bloque (8 bytes para 3DES), se agrega un relleno al final.

   - El relleno consiste en bytes que indican cuántos bytes se agregaron.

Explicación

El cifrado 3DES opera en bloques de 8 bytes (64 bits). Si el texto plano no es un múltiplo de 8, se debe agregar relleno para que tenga la longitud adecuada.

El relleno se realiza de la siguiente manera:

1. Calcula cuántos bytes faltan para que el texto plano sea un múltiplo de 8.
2. Agrega ese número de bytes al final del texto plano.
3. Cada byte de relleno contiene el valor del número de bytes agregados.
4. Por ejemplo, si faltan 3 bytes, se agrega el byte 0x03 tres veces al final del texto plano.

5. Está llenado con el valor 0x03 se elige porque es un byte que representa el número 3 en notación hexadecimal. Cuando se necesita relleno, se agrega un byte con el valor 0x03 tantas veces como sea necesario para completar el bloque.
   EJEMPLOS
   a. Ejemplo 1: Faltan 5 bits (menos de un byte):
      Si faltan 5 bits, eso es menos de un byte.
      En este caso, se agregaría un solo byte de relleno con el valor 0x03 al final del texto.
   b. Ejemplo 2: Faltan 10 bits (1 byte completo):
      Si faltan 10 bits, eso equivale a 1 byte completo.
      Se agregaría un byte de relleno con el valor 0x03 al final del texto.
   c. Ejemplo 3: Faltan 24 bits (3 bytes completos):
      Si faltan 24 bits, eso equivale a 3 bytes completos.
      Se agregarían tres bytes de relleno, cada uno con el valor 0x03.

5. **Cifrado del texto plano**:

   - Se cifra el texto plano utilizando el cifrador 3DES.

   - El resultado es el texto cifrado.

6. **Combinación del IV y el texto cifrado**:

   - Se combina el IV con el texto cifrado.

   - Luego, se codifica todo en base64 para obtener una representación legible y segura.

7. **Guardado del texto cifrado en un archivo**:

   - El texto cifrado codificado en base64 se guarda en un archivo de texto (`cifrado_archivo`).

8. **Mensaje de confirmación**:

   - Finalmente, se imprime un mensaje indicando que el archivo original ha sido cifrado y guardado en el archivo especificado.

TAMAÑO DE LLAVE Y TAMAÑO DE BLOQUE

Tamaño de bloque: 64 bits (8 bytes)

Clave de 192 bits: Utiliza tres claves DES de 64 bits cada una (3 × 64 = 192 bits en total).

Descifrado

1. **Lectura de la clave codificada en base64**:

   - El código abre un archivo que contiene la clave codificada en base64 (`clave_archivo`).

   - Decodifica la clave de base64 para obtener la clave real que se utilizará para el descifrado.

2. **Lectura del texto cifrado codificado en base64**:

   - Abre otro archivo que contiene el texto cifrado codificado en base64 (`cifrado_archivo`).

   - Lee el contenido del archivo y almacena el texto cifrado.

3. **Extracción del IV y el texto cifrado real**:

   - El IV (vector de inicialización) se encuentra en los primeros 8 bytes del texto cifrado.

   - El resto del texto cifrado (después de los primeros 8 bytes) es el texto cifrado real.

4. **Creación del descifrador 3DES en modo CBC**:

   - Se crea un descifrador 3DES en modo Cipher Block Chaining (CBC) utilizando la clave y el IV extraído.

EXPLICACIÓN

El descifrador se crea utilizando la clave 3DES y el vector de inicialización (IV) extraído del texto cifrado. El modo CBC (Cipher Block Chaining) es esencial para descifrar correctamente el texto cifrado.

Funcionamiento

1. Se toma el IV (los primeros 8 bytes del texto cifrado).
2. Se crea un descifrador 3DES en modo CBC utilizando la clave y el IV.
3. El descifrador está listo para procesar el texto cifrado.

5. **Descifrado del texto cifrado**:

   - Se utiliza el descifrador 3DES para descifrar el texto cifrado real.

   - El resultado es el texto plano descifrado.

El proceso es similar al cifrado, pero en sentido inverso:

1. Se toma el texto cifrado real.
2. Se aplica el descifrador 3DES en modo CBC con la clave y el IV.
3. El resultado es el texto plano descifrado.

6. **Eliminación del relleno**:

   - El relleno agregado durante el cifrado se encuentra al final del texto descifrado.

- Se extrae la longitud del relleno (almacenada en el último byte) y se elimina del texto descifrado.

Explicación

1. Se examina el último byte del texto descifrado, que contiene la longitud del relleno agregado.
2. Se extrae esa cantidad de bytes desde el final del texto descifrado.
3. El resultado es el texto plano sin relleno.

7. **Guardado del texto plano descifrado en un archivo**:

- El texto plano descifrado se guarda en un archivo de texto (`descifrado_archivo`).