Ejercicios - Criptografía aplicada

1. Calcular los siguientes hashes de la siguiente frase:

El algoritmo de Diffie-Hellman permite compartir una clave entre 2 personas mientras más personas te están viendo.

Algoritmo	**Hash**
MD5	e879f7ecb97845ab720f0ab3cc8b74d2
SHA-256	8df938d45b4cc88db5d4698aac2aa31b9c437eab2dc67f9c74c6222b51c8580e
SHA-384	9ac0a712cb4701d1453e737f7f74c337ebbc6b2a3a9f1cf6a937ce67022353e2a137077ebbbee0cd5d33cf15475984d3
SHA-512	27ecf7c718481adae4e3633fea96754042e3c5469b1a792dc38fe47413266d922ec7e5773f68eb29b059b1fb0255f5bf4d2741bd466c7228222a12520c9040f6

Realizado con https://emn178.github.io/online-tools/sha512.html

2. Identificar a qué tipo de hashes corresponden los siguientes. ¿Podrá encontrar una entrada que genere cada hash?

Hash	**Algoritmo**	**Una entrada**
e10adc3949ba59abbe56e057f20f883e	MD5	123456
e807f1fcf82d132f9bb018ca6738a19f	MD5	1234567890
c775e7b757ede630cd0aa1113bd102661ab38829ca52a6422ab782862f268646	SHA256	1234567890

https://www.tunnelsup.com/hash-analyzer/
https://hashes.com/es/decrypt/hash

3. Calcular el tiempo necesario para calcular los hashes MD5 de todas las claves entre 6 y 15 caracteres, sabiendo que el tiempo para calcular un hash es de 10 ms.

   3.56x10e29ms → 1.13x10e19 años

4. Desarrollar un programa que permita cifrar y descifrar un texto utilizando AES en modo ECB con una clave de 128, 192 y 256 bits. ¿Qué pasa si el texto a cifrar es 10000 veces la letra A? Si ahora utilizamos AES en modo CBC, ¿qué sucede?

Código en Python

aes.py

```python
from Crypto.Util.Padding import pad, unpad
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from base64 import b64encode, b64decode

key = get_random_bytes(32)
iv = get_random_bytes(16)

def encrypt(plaintext, key):
    cipher = AES.new(key, AES.MODE_CBC, iv)
    return b64encode(cipher.encrypt(pad(plaintext.encode(), 16))).decode()

def decrypt(ciphertext, key):
    cipher = AES.new(key, AES.MODE_CBC, iv)
    return unpad(cipher.decrypt(b64decode(ciphertext.encode())), 16).decode()

def main():
    message = input("Ingrese el mensaje a cifrar: ")
    encrypted = encrypt(message, key)

    print("Mensaje encriptado: ", str(encrypted))
    message_decrypted = decrypt(encrypted, key)
    print(" ")
    print("Mensaje desencriptado: ", str(message_decrypted))

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

Podemos usar también https://asecuritysite.com/Encryption/

Si el texto es siempre el mismo puedo predecir el comportamiento en modo ECB

Si usamos el modo CBC al tener memoria evitamos un ataque de análisis estadístico

5. ¿Recomendaría cifrar el contenido de un sitio web que se envía desde el frontend al backend y viceversa utilizando un algoritmo simétrico? Justifique.

NO ES RECOMENDABLE, ya que un atacante podria obtener nuestra clave del codigo u otro lugar y poder
descrifrar las peticiones. Los algoritmos simetricos se usan para cifrar archivos personales o
sistemas cerrados.

6. Se necesita guardar las claves a través de hashes, ¿cómo haría para que dadas 2 claves iguales, no se guarden con el mismo hash?

Se necesita agregar un string aleatorio que se concatena a la clave antes de calcular el hash.
Este string se denomina salt