Práctica 3

Resultados obtenidos en la práctica:
La clave compartida **KDH = ce44250a450433fe25a75f613ed7aa03.**

Resultado del **Cipher1.txt:**
Received iv: 3ff6d9afad2734d89d5fd046574173f2
Received Cipher: 059100f3e0a256768bdfe4da71ad7a5a2aaca2faf8eb677a079ec8644acc90c7
Received HMAC:
57d6a2d5f95ef68185ba8be33e0b0c4e61fdc1e8675ac8098bec7335d50fdd1b
Calculated HMAC:
906d103e973b3f9c4435a640894b77c1f0dd9f390f932757c6c9b5500efd9efb
Careful! Message has been modified.

Resultado del **Cipher2.txt:**
CIPHER2.TXT:
Received iv: 5d57ce3a1941fa51fdbbf9e7ae9fd535
Received Cipher:
7bd0883165eebcd02d1dd06fe5b59ec636a0d3d0b5caf85542269b6bfd8d14f7
Received HMAC:
46b2ad51353daee9654399954206ba82def4503ee078e61eea003dd232c38dee
Calculated HMAC:
46b2ad51353daee9654399954206ba82def4503ee078e61eea003dd232c38dee
Message is correct!
Decryption key is:
33bf6529dc5165cf4e327220daa9ce25

Por lo tanto la clave k es: **33bf6529dc5165cf4e327220daa9ce25**

Mensaje del top_secret.cipher:
El mensaje del robo de Kutxabank era mentira... Yo soy Satoshi Nakamoto.

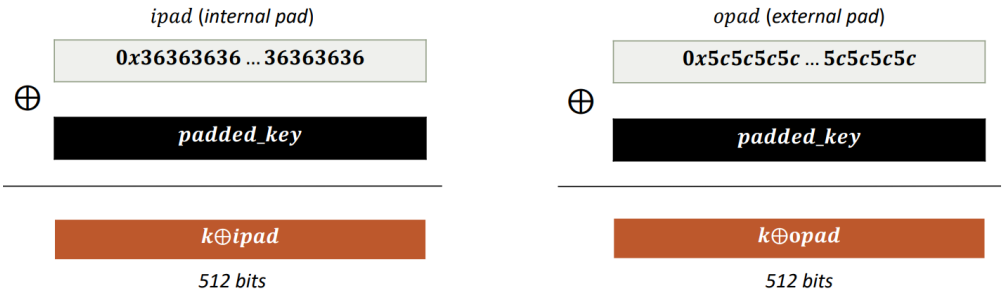
Jose A. Pascual
Primero tenemos que obtener la clave compartida en la carpeta DH-20230216:

1. gcc DH_key_generation.c dhexchange.c -o key
2. ./key

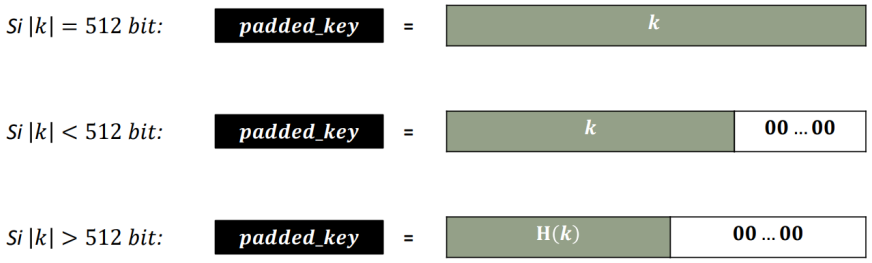
Pasos a seguir para obtener esos resultados en la carpeta Files;

1. ./mac.exe cipher1.txt ce44250a450433fe25a75f613ed7aa03
2. ./mac.exe cipher2.txt ce44250a450433fe25a75f613ed7aa03
3. gcc decrypt_file.c aes.c -o dec
4. ./dec top_secret.cipher destino 33bf6529dc5165cf4e327220daa9ce25

El código documentado en decrypt_MACed_cipher.c:
Lo primero que hemos hecho ha sido declarar ipad y opad, aunque más tarde nos hemos dado cuenta de que ya estaban declarados al principio del código. Vamos a necesitar estas dos funciones para calcular k(+)ipad y k(+)opad:



Una vez hecho eso, hemos asignado el tamaño exacto al padding para luego aplicarlas en las funciones hash1 y hash2.
Padding1 y padding2 serán de 64 bytes, es decir, de 512 bits.
A continuación los hemos inicializado a 0. Una vez hecho esto, con memcpy añadimos en los dos padding la clave que entra como parámetro. Dependiendo del tamaño de key, parse añadirá más ceros, hará la función hash en la clave o simplemente asignará a padding el valor de key, como se indica aquí:



Después tenemos que calcular las funciones hash1 y hash2:
Los tamaños son:

- Para hash1: 64B + tamaño del mensaje
- Para hash2: 64B + tamaño de hash1

Una vez hecho eso, usando las funciones sha256_init, sha256_update y sha256_final calculamos los hashes y obtenemos el resultado de HMAC.

Miembros del grupo:
Ian Fernández y Bianca Ulan.