Universidad del Valle de Guatemala
Cifrado de Información - CC 3078 - Sección 10
Julio Herrera 19402
Bryann Alfaro 19372
Diego Arredondo 19422

Intercambio de claves Método Diffie-Hellman
Laboratorio 8

INTRODUCCIÓN

En el presente laboratorio se buscará comprender el funcionamiento del método Diffie-Hellman para el intercambio de claves. Se procederá a utilizar este algoritmo usando las fórmulas matemáticas correspondientes para poder verificar que al final del proceso, las claves compartidas sean iguales. Este proceso se llevará a cabo con diferente longitud de primos, específicamente de 6, 10 y 14 dígitos.

Seguidamente, se procederá a utilizar la librería pyDH para realizar el proceso de intercambio por medio de la colaboración con otro grupo de trabajo y compartir mensajes cifrados utilizando la clave compartida generada. En este caso, se utilizará el método AES SHA-256 para la encripción y un group de 15 para la generación de claves.

METODOLOGÍA

Primera parte:

Datos

1. Número primo: 57
2. Número primo de 6 dígitos: 131071
3. Número primo de 10 dígitos: 1190494759
4. Número primo de 14 dígitos: 67280421310721

Para esta parte se hará un intercambio de claves utilizando el método de Diffie-Hellman. Primeramente se deben de fijar 2 valores:

- Un número primo p > 2
- Un generador g donde 1 < g < p - 1
-

Luego de esto, se simulará el intercambio de claves entre dos participantes.
los pasos que seguirán son:

- Generar una clave secreta para cada uno de los participantes.
- El participante “A” debe enviar el valor g^a
- El participante “B” debe enviar el valor g^b

- Luego, el participante “A” debe calcular B^{a}, mientras que el participante “B” debe calcular A^{b}.
- Verificar si los resultados coinciden.

Segunda parte:

La segunda parte de este laboratorio consiste en generar una conversación con otro equipo de trabajo, utilizando la generación de claves con el método de Diffie-Hellman.

1. Como primer paso se definió el método de generación de claves de Diffie-Hellman como el 15 (3072 bit) y el método de encriptación de los mensajes será AES en modo CBC con SHA-256.

Durante el desarrollo del laboratorio, cada equipo realizó su propia implementación funcional de generación de claves compartidas, así como de cifrado y descifrado de mensajes, sin embargo al hacer la reunión se tuvieron que realizar cambios de ambos lados para cifrar y descifrar de forma conveniente según el objetivo del laboratorio (crear una conversación en tiempo real).

2. Primero se genera d_1 la cual es nuestra llave privada y que a partir de ella generamos nuestra llave pública $d_{1\,pub}$.
3. Luego recibimos la llave pública del otro equipo como input en el programa y enviamos la nuestra a ellos, para ambos generar la llave compartida a partir de las llaves privadas, en nuestro caso, d_1.

Definido el método de cifrado de mensajes como AES en modo CBC, recordemos que este requiere de una llave la cuál será la clave compartida que ambos equipos generamos y así poder cifrar y descifrar nuestros mensajes solo entre nosotros. Sin embargo esta llave debe tener cierto tamaño en específico, en este caso 256 bits y es por ello que pasaremos la clave compartida por un hash de SHA-256 antes de utilizarla como llave en AES.

4. Convertimos la clave compartida que es un número, a un texto plano utilizando codificación utf-8, luego se obtiene el resultado de este hash como un binario de 256 bits (digest) el cuál es el que se utilizará en AES.

Aquí es donde ambos equipos adaptamos nuestros métodos, principalmente para simplificar el envío y recepción de mensajes, concatenando el bloque de inicialización al texto cifrado, para así enviar solo un string que contiene ambos elementos y separarlos al descifrar.

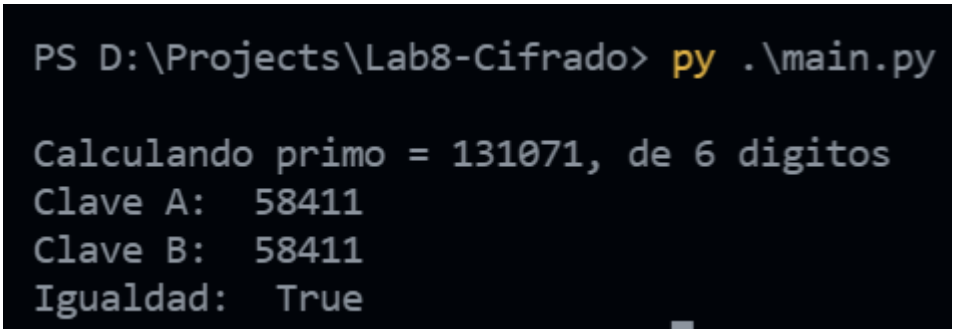
5. Cifrado de mensajes
    a. Se recibe el mensaje a cifrar.
    b. Se genera un bloque aleatorio de inicialización (IV) de 16 bits.
    c. Se genera el cifrado AES utilizando como llave el digest, el modo CBC y el bloque IV.
    d. Se cifra el texto (*cypher text*) con un relleno (pad) de 16 bits.
    e. Se concatena el bloque IV con el texto cifrado y se codifican a base 64, luego se decodifican en utf-8 para poder ser enviados como un solo string.
6. Descifrado de mensajes
    a. Se recibe el string a descifrar.

b. Se genera el cifrado AES utilizando como llave el digest, el modo CBC y como bloque IV se utilizan los primeros 16 bits del string a descifrar decodificados en base64.
c. El mensaje descifrado (*plain text*) se obtiene de desencriptar solo los bits restantes del string, es decir de los primeros 16 bits en adelante y quitar el relleno (unpad) de 16 bits.

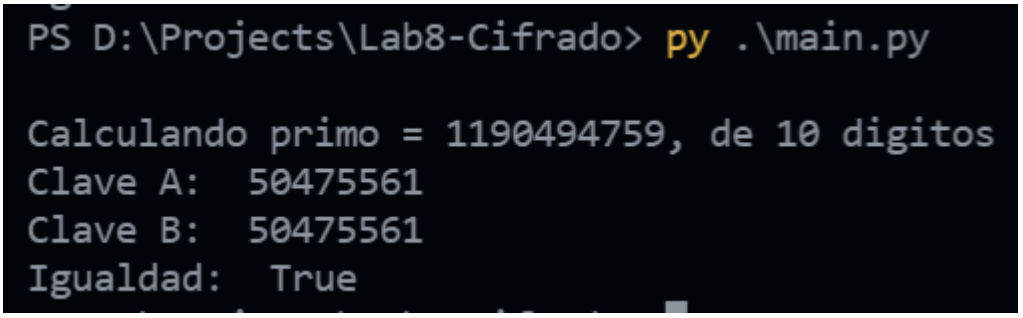
RESULTADOS

Primera parte:

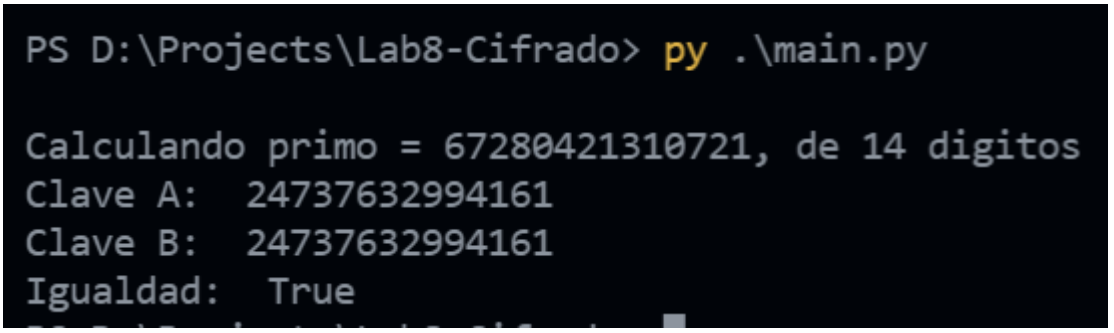
Primo 6 dígitos



Primo 10 dígitos



Primo 14 dígitos

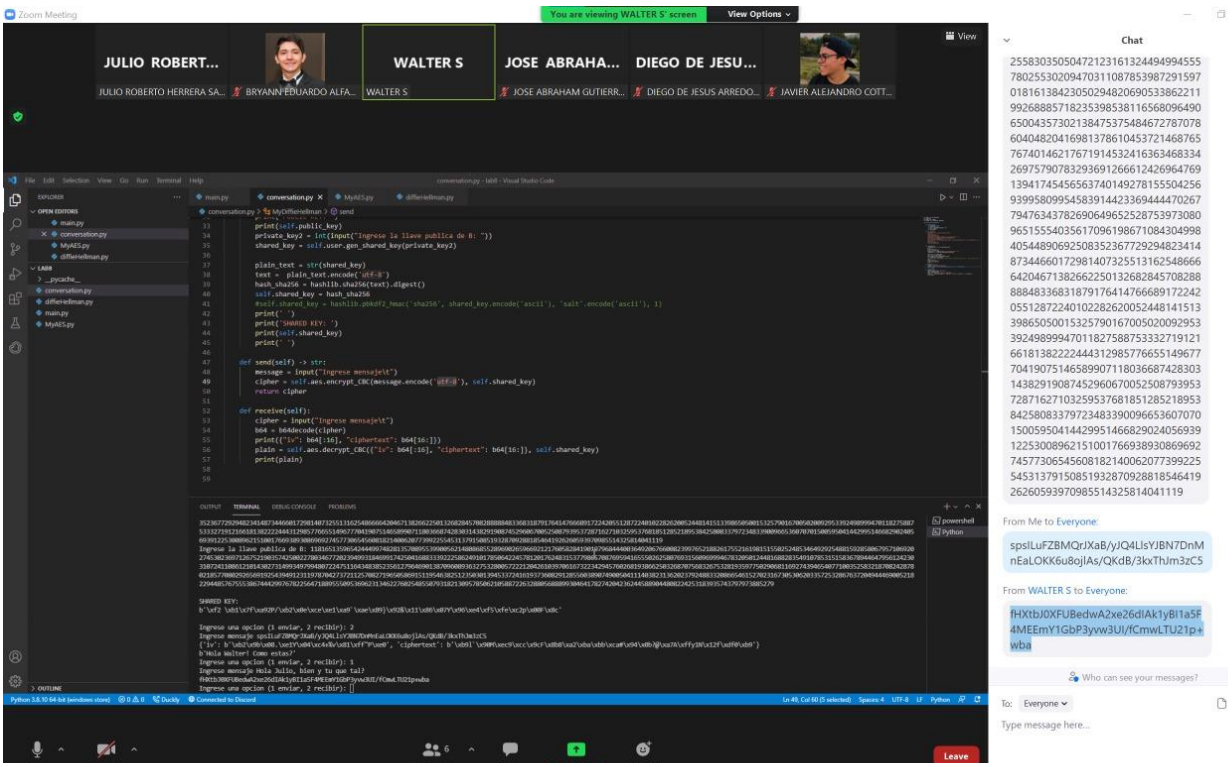


Segunda parte:

```
SHARED KEY:
b'&}Sg\xce\xb3\x9ba\x91~\x8f\xc8\x12\x90\x01\x95\x00\xf4\x82<\xa3QZ\x82\xaa\xe1\x80S\x81U\t\x89'
Ingrese mensaje 4QXDJyXxPXcqr347325JchmDNuMQqbJbIbHHP3C6Dp2HC2ehrLJLeBT7QXmWX5bkf1HK7TpbyXuM0hCpdUUcUQ==
{'iv': b"\xe1\x05\xc3'%\xf1~w*\xaf~;\xdd\x9eIr", 'ciphertext': b'\x19\x836\xe3\x10\xa9\xb2[!\xb8G?p\xba\x0e\
b'Hola! este es un mensaje de Julio'
PS C:\Users\Danilo\Desktop\UVG\6to Semestre\Cifrado\lab8>
```

```
Clave compartida:  8e3b6e396420f8587657927efc6faca04c545128b19ca3c81678463833730c77
Binario:        b'\xf2 \xb1\x7f\xa92P/\xb2\x0e\xce\xe1\xa9`\xae\x89}\x92&\x11\x86\x07Y\x96\xe4\xf5\xfe\xc2p\x
enviar
Hola Walter! Como estas?
Mensaje original:  Hola Walter! Como estas?
Mensaje cifrado:  spslLuFZBMQrJXaB/yJQ4LlsYJ8N7DnMnEaLOKK6u8ojlAs/QKdB/3kxThJm3zC5
recibir
fHXtbJ0XFUBedwA2xe26dIAk1yBI1a5F4MEEmY1GbP3yvw3UI/fCmwLTU21p+wba
b'|u\xedl\x9d\xd7\x15@`w\x006\xc5\xed\xbat'
b'\x80$\xd7 H\xd5\xaeE\xe0\xc1\x04\x99\x8dFl\xfd\xf2\xbf\r\xd4#\xf7\xc2\x9b\x02\xd35m1\xfb\x06\xda'
Mensaje recuperado:  b'Hola Julio, bien y tu que tal?'
```

Evidencia de la Reunión:



CONCLUSIONES

1. Dados los resultados, se puede concluir que el método de Diffie-Hellman es muy eficiente para el intercambio de claves y mensajes de dos orígenes distintos, ya que deben contar ambas partes con la clave pública y generar la misma clave compartida, eso hace muy seguro dicho método.

2. En la parte 1 se observó que no existe gran diferencia de tiempo en el intercambio de claves, por lo que se induce a utilizar primos con mayor cantidad de dígitos para aumentar la seguridad del método.
3. Es importante ir cambiando las claves privadas y públicas (y por ende las compartidas) cada cierto tiempo y por ello la conversación se hace en tiempo real para que así cada vez sea un nuevo par de claves que no puedan ser encontradas.