Juan Pablo Rivera - 202211439

Sofia Losada - 202221008

Infraestructura Computacional

Documentación Caso 3

La solución propuesta, está construida por tres clases. La clase Servidor; en donde se realiza la conexión al servidor, la clase Cliente y la clase Delegado. A continuación, se da una descripción de en donde se implementan los siguientes métodos:

1. Cliente:
    a. Método main: En este método inicialmente se declaran los sockets: escritor de tipo PrintWritter y lector de tipo BufferReader. Se cargan las llaves públicas y privadas, y se generan los valores de g,p y Gx.
    b. Metodo loadPublicKey
    c. Método crearKey SecretaCompartida
    d. Método deriveKeys: tiene por parámetro la llave secreta compartida y convierte el parámetro una lista de 32 bytes. Crea una lista de 32 bytes “hmacKeyBytes” . Crea la llave simétrica
    e. enviarDatosCifrados: tiene como parametros UID, package_id, autor, SecretKey symmetricKey, SecretKey hmacKey. El servidor recibe y cifra datos (UID y package_id) junto con sus HMAC. Implementa AES en formato CBC para encriptación del UID y del package_id, para el HMAC de cada datos encriptados. Transmite al servidor el vector de inicialización (IV), los datos encriptados y sus HMAC.
    f. RecibiryVerificarEstado: tiene como parámetros lector, IvParameterSpec ivSpec, SecretKey symmetricKey, SecretKey hmacKey). Obtiene, descifra y comprueba la integridad del estado del paquete transmitido desde el servidor.Se le proporciona el estado encriptado y su HMAC.Finalmente, erypta el estado utilizando AES en modo CB
2. Delegado:
    a. enviarCifrados de Datos: tiene como parametros UID, package_id, autor, SecretKey symmetricKey, SecretKey hmacKey)
    b. El servidor recibe y cifra datos (UID y package_id) junto con sus HMAC.
    c. Hacer uso de AES en formato CBC para encriptación del UID y del package_id. Determina el HMAC de cada datos encriptados.
    d. Transmite al servidor el vector de inicialización (IV), los datos encriptados y sus HMAC.
    e. YVerificarEstado: tiene como parámetros lector, IvParameterSpec ivSpec, SecretKey symmetricKey, SecretKey hmacKey). Obtiene, descifra y comprueba la integridad del estado del paquete transmitido desde el servidor. Se le proporciona el estado encriptado y su HMAC, encripta el estado utilizando AES en modo CBC. Además,

comprueba la veracidad del estado al comparar el HMAC obtenido con el HMAC estimado.Si la verificación resulta exitosa, imprime el estado del paquete.

Los demás métodos son visibles en el código.

3. Especificaciones para correr el programa y consideraciones:

Para que el programa se ejecute correctamente se debe correr primero la clase Servidor, luego se debe correr Cliente.

El código genera las llaves a partir de archivos, los cuales se incluyen en la carpeta de entrega. Para que se generen correctamente se debe cambiar la ruta de archivo. En la clase Cliente se debe cambiar en el método loadPublicKey. Para esto debe dar click derecho sobre el archivo public.key ----> Copy path ------> actualizar la función loadPublicKey donde se encuentre la ruta de archivo anterior. En la clase Delegado se debe cambiar el método loadPrivateKey. Para esto debe dar click derecho sobre el archivo private.key ------> Copy path -------> actualizar la función loadPrivateKey donde se encuentra la ruta de archivo anterior.

En la implementación para crear los valores de g y p, se intentó implementar ssl, para que más casos de prueba fueran aceptados. Sin embargo en el desarrollo de la implementación surgió el siguiente error:

at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1170)

at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1089)

at java.base/java.lang.Runtime.exec(Runtime.java:681)

at java.base/java.lang.Runtime.exec(Runtime.java:491)

at java.base/java.lang.Runtime.exec(Runtime.java:366)

at Delegado.handshake(Delegado.java:176)

at Delegado.run(Delegado.java:53)

at java.base/java.lang.Thread.run(Thread.java:1583)

Caused by: java.io.IOException: CreateProcess error=5, Acceso denegado

at java.base/java.lang.ProcessImpl.create(Native Method)

at java.base/java.lang.ProcessImpl.<init>(ProcessImpl.java:500)

at java.base/java.lang.ProcessImpl.start(ProcessImpl.java:159)

at java.base/java.lang.ProcessBuilder.start(ProcessBuilder.java:1126)

... 7 more

Si bien seguimos todos los pasos para descargar y ejecutar el algoritmo, el error mostró que no se tenían los permisos disponibles para implementar la librería, por lo que no fue posible crear los valores utilizándolo. De igual manera, la implementación realizada funciona

IMPORTANTE!:

Así mismo, para generar el reto se debe poner explícitamente su valor en cliente.

Por el mismo error explicado anteriormente, el p y el g no se pueden generar, por lo que es necesario que en delegado se pongan explícitamente sus valores

4. Escenario i, Cliente hace 32 peticiones

CASO I, UN CLIENTE MANDA 32 SOLICITUDES	DESENCRIPTAR RETO	52
	GENERAR VARIABLES	31
	C1	3
	C2	7
	C3	6
	C4	6
	C5	7
	C6	3
	C7	8
	C8	3
	C9	7
	C10	3
	C11	8
	C12	6
	C13	1
	C14	5
	C15	8
	C16	2
	C17	3
	C18	5
	C19	7
	C20	7
	C21	3
	C22	2
	C23	7
	C24	7
	C25	8
	C26	3
	C27	7
	C28	2
	C29	3
	C30	6
	C31	2
	C32	5

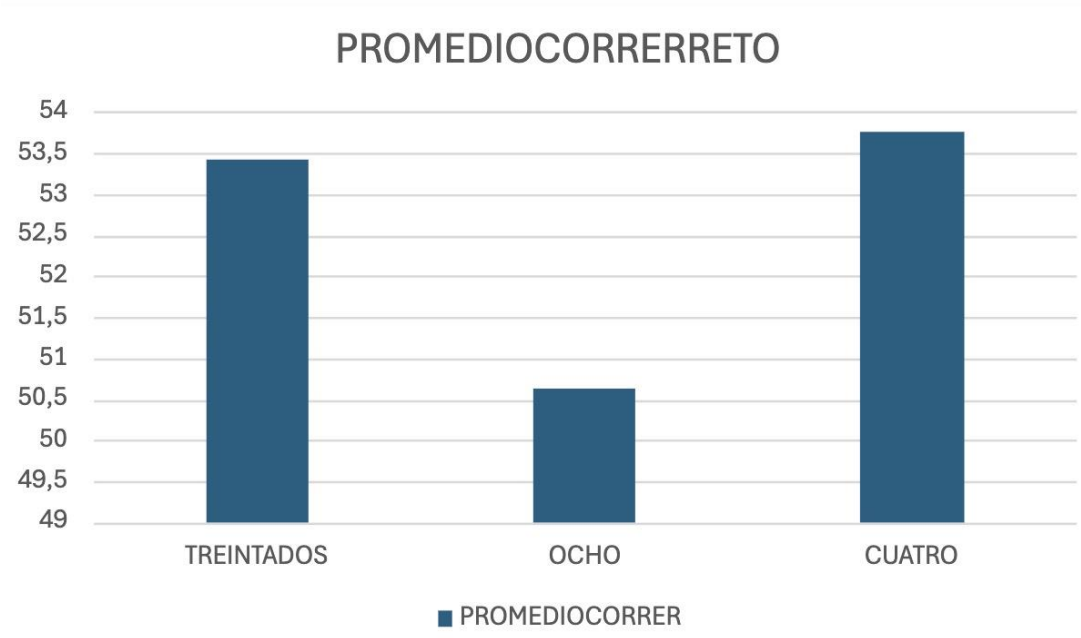
5. Escenario ii, 32 Clientes hacen una petición

		RETO	VARIABLES	CONSULTA
CASO II, 32 CLIENTES MANDAN 1 SOLICITUD	C1	35	23	1
	C2	63	31	4
	C3	37	23	3
	C4	43	28	6
	C5	54	25	3
	C6	51	32	6
	C7	44	27	1
	C8	49	32	2
	C9	39	32	4
	C10	47	25	6
	C11	46	20	3
	C12	64	23	5
	C13	41	30	1
	C14	60	30	5
	C15	63	25	6
	C16	58	23	4
	C17	66	20	3
	C18	64	27	6
	C19	64	26	2
	C20	35	30	1
	C21	46	23	7
	C22	46	32	5
	C23	37	28	4
	C24	49	29	8
	C25	70	31	3
	C26	51	27	7
	C27	49	22	4
	C28	51	27	2
	C29	68	25	6
	C30	36	21	4
	C31	49	31	8
	C32	48	22	8

		RETO	VARIABLES	CONSULTA
CASO II, 8 CLIENTES MANDAN 1 SOLICITUD	C1	68	22	7
	C2	64	30	2
	C3	61	29	7
	C4	69	21	7
	C5	37	32	5
	C6	38	24	3
	C7	61	28	5
	C8	54	32	5

CASO II, 4 CLIENTES MANDAN 1 SOLICITUD				
		RETO	VARIABLES	CONSULTA
	C1	50	21	5
	C2	71	31	1
	C3	53	24	1
	C4	44	31	6

GRAFICA 1:



GRAFICA 3:

TIPO	PROMEDIOCORRER
E SIMETRICO	0.67
DE SIMETRICO	2.3
E ASIMETRICO	0.1
DE ASIMETRICO	0.06

GRAFICA 4:

TIPO	PROMEDIOCORRER
CUATRO	3.1
OCHO	2.3
TREINTA Y DOS	2.1