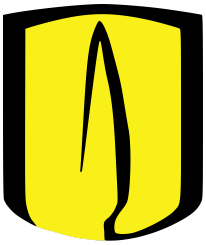
Universidad de los Andes



Infraestructura computacional

2024-1

Caso de Estudio 3 – Canales Seguros

Juan David Orduz - 202123170
Henry Santiago Antolinez - 202121785
Raúl Santiago Rincón - 202120414

1. Descripción de la organización de los archivos en el zip.

- docs; acá se ubican los archivos de documentación del caso y todos los tiempos medidos y graficados.
- src; en esta se encuentra el código que se divide en 9 clases (archivos .java)

2. Instrucciones para correr el servidor y el cliente, incluyendo cómo configurar el número de clientes concurrentes

Tanto el Servidor como el Cliente tienen sus propias clases y cada una posee un método main[], por lo que para ejecutarlos simplemente se debe ejecutar cada archivo de forma independiente. Sin embargo, es crucial seguir un orden específico: primero se debe ejecutar el Servidor y luego el Cliente.

Si se desea ejecutar más de un Cliente simultáneamente, se puede utilizar el archivo "multiplesClientes.java". Al ejecutar este archivo, se solicitará al usuario la cantidad de clientes que se desean ejecutar a través de la consola. Posteriormente, se debe especificar el número deseado y automáticamente se ejecutarán los clientes en paralelo.

3. Preguntas

a. En el protocolo descrito el cliente conoce la llave pública del servidor (K_w+). ¿Cuál es el método comúnmente usado para obtener estas llaves públicas para comunicarse con servidores web?

El método comúnmente utilizado para obtener la clave pública de un servidor web es a través del protocolo SSL/TLS. Cuando un cliente se conecta a un servidor web a través de HTTPS (HTTP Secure), el servidor presenta su certificado digital, que contiene su clave pública. El cliente luego verifica este certificado para asegurarse de que es válido y confiable, y extrae la clave pública del servidor para establecer una comunicación segura. Este proceso de verificación se realiza utilizando una infraestructura de clave pública (PKI) que incluye autoridades de certificación (CA) que emiten y certifican los certificados digitales de los servidores. En el caso de los servidores web este protocolo implementa criptografía asimétrica y comúnmente se le conoce como certificado. El certificado SSL/TLS de un sitio web se comparte públicamente ya que contiene la llave pública. Cuando un cliente desea establecer una conexión segura con un servidor, el servidor presenta su certificado SSL/TLS, que incluye la llave pública, el cliente puede entonces usar esta clave pública para encriptar la información que se enviará al servidor.

b. ¿Por qué es necesario cifrar G y P con la llave privada?

Es necesario cifrar G y P (el valor generado aleatoriamente) con la llave privada del servidor para garantizar la autenticidad del servidor y evitar ataques de suplantación de identidad (man-in-the-middle). Al cifrar G y P con la llave privada, es posible verificar que solo el servidor legítimo que posee esa llave privada puede haber generado ese cifrado. Cuando el cliente recibe el cifrado de G y P, puede descifrarlo con la llave pública del servidor y verificar que efectivamente proviene del servidor legítimo. De esta manera se establece una conexión segura y auténtica con el servidor antes de intercambiar más información que podría ser sensible.

c. El protocolo Diffie-Hellman garantiza “Forward Secrecy”, presente un caso en el contexto del sistema Banner de la Universidad donde sería útil tener esta garantía, justifique su respuesta (por qué es útil en ese caso).

Un caso relevante para aplicar la garantía de "Forward Secrecy" en el sistema Banner de la Universidad de los Andes es la protección de la información confidencial de los estudiantes, como notas, expedientes académicos y datos financieros.

Esta propiedad asegura que, aunque la clave privada del servidor o cliente sea comprometida en el futuro, por ejemplo, debido a un ciberataque exitoso o una filtración interna, los mensajes cifrados previamente no podrán ser descifrados por un atacante. A pesar de las medidas de seguridad implementadas, siempre existe el riesgo de que una clave privada sea vulnerada.

Sin Forward Secrecy, si un ciberdelincuente obtiene la clave privada, podría descifrar todos los mensajes anteriores protegidos con esa clave, accediendo así a información histórica confidencial de los estudiantes, lo cual representa un grave riesgo para su privacidad.

Al utilizar Diffie-Hellman con Forward Secrecy, cada sesión de comunicación emplea una clave temporal única que se descarta al finalizar. Por lo tanto, aunque la clave privada sea revelada posteriormente, los mensajes previos permanecerán seguros e inaccesibles, resguardando la confidencialidad y privacidad de los datos estudiantiles en el sistema Banner a largo plazo.

4. Toma de datos del Cliente:

a. Tiempo del cliente para verificar la firma (Promedio)

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	5.25	2.125	1.34375

b. Tiempo del cliente para calcular Gy

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	0	0	0

c. Tiempo del cliente para cifrar la consulta

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	0	0.0625	0.125

d. Tiempo del cliente para generar el código de autenticación

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	4.25	3.1875	3.21875

e. Tiempo del servidor en verificar el código de autenticación

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	0.75	0.5625	0.4375

5. Toma de datos del Servidor:

a. Tiempo del servidor para generar la firma

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	9.5	3.3125	2.03125

b. Tiempo del servidor en descifrar la consulta

	Número de clientes		
	4	**16**	**32**
Tiempo en ms	0.75	0.1875	0.34375

6. Tabla con los datos recopilados y gráfica con los datos de la tabla.

a. Tabla de clientes

			Tiempos en ms				
			Calcular G^y	Cifrar consulta	Verificar Código de Autenticación	Generar código de Autenticación	Verificar firma
Número de clientes	4	C1	0	0	1	5	17
		C2	0	0	0	4	1
		C3	0	0	1	5	1
		C4	0	0	1	3	2
	16	C1	0	0	1	2	19
		C2	0	0	1	4	2
		C3	0	0	1	4	1
		C4	0	0	0	3	1
		C5	0	0	1	3	0
		C6	0	1	0	3	2
		C7	0	0	1	3	1
		C8	0	0	0	3	1
		C9	0	0	0	3	2
		C10	0	0	1	3	0
		C11	0	0	0	3	0
		C12	0	0	1	3	1
		C13	0	0	0	4	1
		C14	0	0	1	3	1
		C15	0	0	1	3	1
		C16	0	0	0	4	1
	32	C1	0	0	0	4	18
		C2	0	1	1	4	1
		C3	0	0	1	4	1
		C4	0	0	1	3	1
		C5	0	1	1	3	1
		C6	0	1	1	4	1
		C7	0	0	0	3	1
		C8	0	0	1	3	1
		C9	0	0	1	3	1

		C10	0	0	0	3	0
		C11	0	0	1	3	1
		C12	0	0	0	3	1
		C13	0	1	1	3	1
		C14	0	0	0	3	1
		C15	0	0	0	3	1
		C16	0	0	1	3	1
		C17	0	0	0	3	0
		C18	0	0	1	3	1
		C19	0	0	0	4	1
		C20	0	0	0	5	0
		C21	0	0	0	3	1
		C22	0	0	0	4	1
		C23	0	0	1	3	1
		C24	0	0	0	3	0
		C25	0	0	0	3	1
		C26	0	0	1	3	1
		C27	0	0	0	3	0
		C28	0	0	1	3	1
		C29	0	0	0	3	1
		C30	0	0	0	3	1
		C31	0	0	0	3	0
		C32	0	0	0	2	1

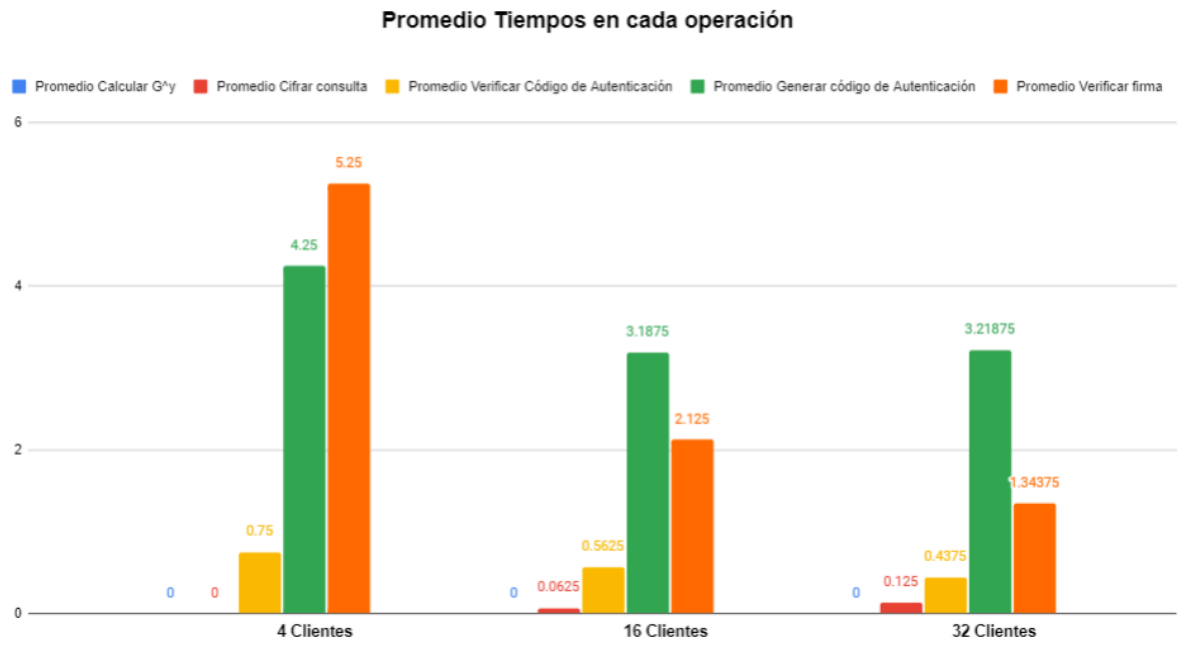
b. Tabla Servidor

			Tiempos en ms	
			Descifrar Consulta	**Generar firma**
Número de clientes	**4**	**C1**	1	35
		C2	1	1
		C3	1	1
		C4	0	1
	16	**C1**	0	34

		C2	0	1
		C3	0	1
		C4	1	1
		C5	0	0
		C6	1	0
		C7	0	0
		C8	0	0
		C9	0	0
		C10	0	1
		C11	0	10
		C12	1	1
		C13	0	1
		C14	0	1
		C15	0	1
		C16	0	1
	32	**C1**	1	40
		C2	0	2
		C3	0	1
		C4	0	1
		C5	0	0
		C6	1	1
		C7	0	1
		C8	0	1
		C9	0	1
		C10	1	1
		C11	0	1
		C12	1	1
		C13	1	0
		C14	0	1
		C15	0	1
		C16	0	1
		C17	0	1
		C18	0	1
		C19	1	0

		C20	1	1
		C21	0	1
		C22	1	1
		C23	1	1
		C24	0	1
		C25	0	0
		C26	1	0
		C27	0	1
		C28	0	0
		C29	0	1
		C30	0	1
		C31	1	0
		C32	0	1

Gráfica de tiempos promedio par los cálculos hechos por los clientes:



Gráfica de tiempos promedio par los cálculos hechos por los servidores delegados:

7. Comentarios sobre los resultados.

Los tiempos de ejecución para las operaciones criptográficas en el cliente muestran algunas tendencias interesantes. El tiempo para calcular G^y es prácticamente despreciable en todos los escenarios, lo que indica que esta operación no representa un cuello de botella significativo. Sin embargo, el tiempo para cifrar la consulta, aunque bajo, muestra un ligero incremento a medida que aumenta el número de clientes, lo que sugiere que el proceso de cifrado podría convertirse en un cuello de botella en escenarios con una cantidad aún mayor de clientes concurrentes. El tiempo para generar el código de autenticación (HMAC) se mantiene relativamente constante y no se ve afectado significativamente por el número de clientes.

En el lado del servidor, el tiempo para generar la firma digital es el más significativo y muestra una disminución notable a medida que aumenta el número de clientes. Este comportamiento podría atribuirse a la optimización de las operaciones criptográficas o a una mejor distribución de la carga de trabajo entre los hilos del servidor. El tiempo para descifrar la consulta es relativamente bajo y no parece verse afectado significativamente por el número de clientes.

En general, las operaciones criptográficas, como la generación y verificación de firmas digitales, parecen ser las más costosas en términos de tiempo de ejecución, tanto en el lado del cliente como en el del servidor. Sin embargo, a medida que aumenta el número de clientes, los tiempos de ejecución no se incrementan de manera proporcional, lo que sugiere que el sistema puede escalar razonablemente bien con un mayor número de clientes concurrentes. Es

importante tener en cuenta que los tiempos de ejecución pueden verse afectados por varios factores, como la carga de trabajo del sistema, la configuración de hardware y software, y la implementación específica de los algoritmos criptográficos. Para obtener un análisis más completo y detallado, sería recomendable realizar pruebas adicionales con una variedad más amplia de escenarios y cargas de trabajo, así como examinar otros aspectos del rendimiento, como el uso de memoria, el consumo de CPU y la utilización de red.

8. Estimación de cuántas consultas puede cifrar su máquina, cuántos códigos de autenticación puede calcular y cuántas verificaciones de firma, por segundo.

Caracteristicas del PC utilizado para las pruebas:

- AMD Ryzen 7 5800X, 3.8 GHz
- 16 GB RAM

	promedio en ms	Numero de operaciones	ms x n de operaciones
cifrar consulta	0.125	8000	1000
codigos de autenticacion	3.65625	275	1005.46875
verificaciones de firma	1.34375	745	1001.09375

En esta tabla podemos ver que para esta máquina y su velocidad, estimadamente podemos

Realizar 8000 operaciones de cifrado de consulta por segundo, 275 códigos de autenticación calculados, entre generados y verificados. y por último 745 verificaciones de firma en 1 segundo

9. Referencias

(1) ¿Cómo funciona la criptografía de clave pública? - Cloudflare. https://www.cloudflare.com/es-es/learning/ssl/how-does-public-key-encryption-work/.

(2) Configurar claves SSH públicas y privadas | www.isholgueras.com. https://www.isholgueras.com/blog/configurar-claves-ssh-publicas-y-privadas.

(3) SSH Key: guía completa de las llaves públicas y privadas de SSH. http://codigoelectronica.com/blog/ssh-key-guia-completa-de-las-llaves-publicas-y-privadas-de-ssh.
(4) Utiliza tus llaves para conectarse a un servidor vía SSH. https://lamiradadelreplicante.com/2015/01/02/utiliza-tus-llaves-para-conectarse-a-un-servidor-via-ssh/
(5) Diffie-Hellman and Forward Secrecy. https://www.zwilnik.com/security-and-privacy/diffie-hellman-and-forward-secrecy/
(6) Perfect Forward Secrecy (PFS). https://avinetworks.com/glossary/perfect-forward-secrecy/