Mariana Díaz Arenas- 202020993

Daniel Osorio Cárdenas - 202022996

Caso 3 – Informe

En el protocolo descrito el cliente conoce la llave pública del servidor (K_w+). ¿Cuál es la manera común de enviar estas llaves para comunicaciones con servidores web?

Para poder enviar llaves públicas para comunicaciones con servidores web se pueden usar certificados digitales, pues estos documentos buscan identificar a una persona con su respectiva llave pública para permitir realizar transacciones con un servidor, permitiendo ejecutar procedimiento como el descrito para el caso. Durante un intercambio TLS, que es la tecnología sobre la que se basa HTTP(s), el servidor envía su certificado al cliente a través de TCP. Este certificado está firmado por una autoridad en la que los clientes confían y contiene la clave pública del servidor.

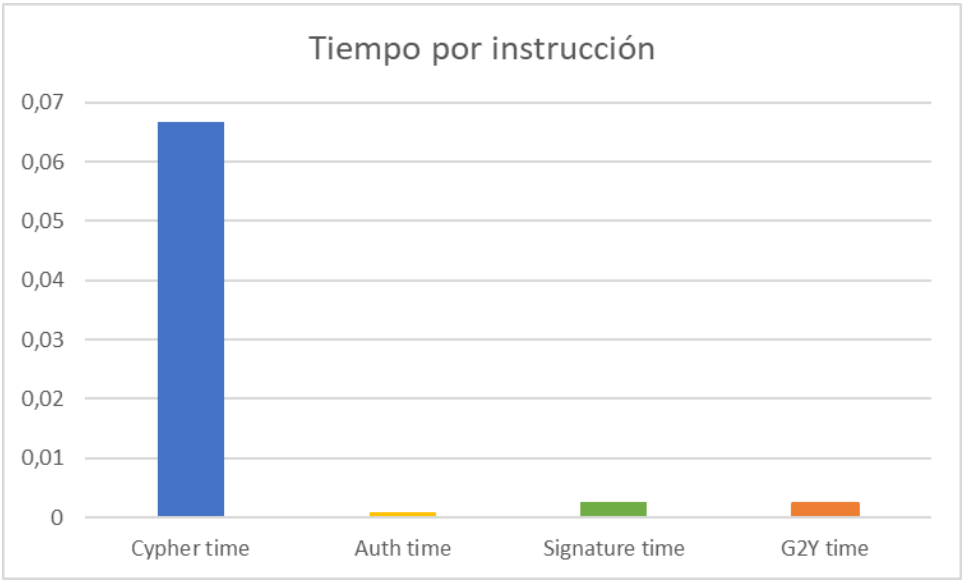
El protocolo Diffie-Hellman garantiza "Forward Secrecy", explique en qué consiste esta garantía.

Forward secrecy garantiza que si llegase el caso de un ataque que descubre las llaves utilizadas actualmente, no va a afectar la seguridad de los datos generados anteriormente o a un futuro, pues busca generar una llave diferente para cada ejecución del programa para no poner en riesgo la seguridad de todo el sistema. Diffie Helman asegura esto pues, cada vez que se ejecuta el programa, se está generando un nuevo valor de x y y, creando una llave simétrica diferente a las utilizadas anteriormente, por lo que se está asegurando que se está utilizando una nueva llave y a pesar de que ocurriese un ataque con un determinado evento, no va a afectar la seguridad de los otros del sistema.

Tabla de datos recopilados

En la siguiente tabla se muestra el tiempo promedio (en segundos) que tomó hacer las diferentes operaciones mostradas en la tabla con una cantidad de usuarios diferentes. Es importante tener en cuenta, que no en todos los casos se ejecutan todas las operaciones debido al funcionamiento del servidor.

Cantidad de clientes	*Cifrar la consulta*	*Generar código de autenticación*	*Verificar la firma*	*Calcular G^y*
4	0.06	0.001	0.003	0.002
16	0.06	0.001	0.002	0.003
32	0.08	0.001	0.003	0.002



A partir de la tabla de datos y la gráfica generada a partir de ellos, se puede observar que la instrucción de cifrar la consulta es lo que más tiempo toma y generar la autenticación es la instrucción que menos toma tiempo. A pesar de ello, se puede evidenciar que son tiempos muy cortos que tarda el programa en atender a un cliente y que, por efectos de la concurrencia, sin importar la cantidad de clientes que ingresen al sistema, las operaciones van a continuar ejecutándose en el mismo tiempo.

Velocidad del procesador y estimación de tiempos

Procesador: AMD Ryzen 5 5500U, 6 núcleos, 12 hilos, con velocidad base de reloj de 2.10 GHz

Cifrar consulta:

$$promedio = \frac{0.06 + 0.06 + 0.08}{3} = 0.066\ segundos\ por\ cifrado$$

Entonces, si tenemos que un cifrado se demora 0.066 segundos, entonces en un segundo tendríamos:

$$\frac{0.066s}{1\ cifrado} = \frac{1s}{y\ cifrados}$$

Despejando y:

$$y = \frac{1}{0.066} = 15.15 \approx 15\ cifrados\ por\ segundo$$

Siguiendo el mismo procedimiento para las otras instrucciones tenemos:

Generar código de autenticación:

$$promedio = 0.001\ segundos\ por\ código\ de\ autenticación$$

Entonces:

$$y = \frac{1}{0.001} = 1000\ códigos\ de\ autenticación\ por\ segundo$$

Verificación de la firma:

$$promedio = \frac{0.003 + 0.002 + 0.003}{3} = 0.0026\ segundos\ por\ código\ de\ autenticación$$

Entonces:

$$y = \frac{1}{0.0026} = 384.6 \approx 384\ verificaciones\ de\ firma\ por\ segundo$$

Calcular G^y:

$$promedio = \frac{0.002 + 0.003 + 0.002}{3} = 0.0023\ cálculos\ de\ G^y\ por\ segundo$$

Entonces:

$$y = \frac{1}{0.0023} = 428.57 \approx 428\ cálculos\ de\ G^y\ por\ segundo$$

Aclaraciones sobre el programa

Para ejecutar el programa, se debe correr primero la clase de ServidorMain y después la de ClienteMain donde se verá por consola, respectivos prints que permiten verificar el paso a paso del proceso de intercambio de información. En el cliente, se realiza un print que permite verificar si se ha hecho el procedimiento de agregarle un 1 al número que se envía correctamente, dependiendo de la opción que se ejecute en el servidor.

*Para cambiar la cantidad de clientes que se mandan al servidor, se debe cambiar el límite del for que están en el main de la clase ClienteMain

Referencias

- Lesson: All About Sockets (The JavaTM Tutorials > Custom Networking). (s. f.). https://docs.oracle.com/javase/tutorial/networking/sockets/index.html
- *Diffie-Hellmann Overview*. (s. f.). https://www.ibm.com/docs/en/zvse/6.2?topic=SSB27H_6.2.0/fa2ti_openssl_consider_diffie_hellman.htm
- *Diffie-Hellman and Forward Secrecy*. (2020, 6 septiembre). zwilnik. https://www.zwilnik.com/security-and-privacy/diffie-hellman-and-forward-secrecy/