Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
Semestre 2024-20

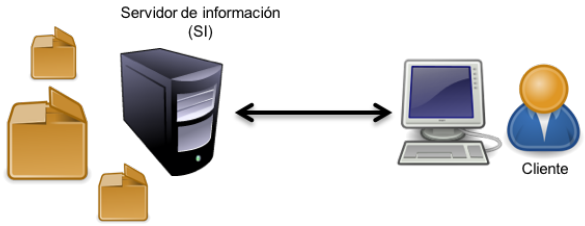
Caso de Estudio 3 – Canales Seguros
Sistema de rastreo de paquetes en una compañía transportadora

Objetivos

- Comprender ventajas y limitaciones de los algoritmos para cifrar datos.
- Construir un prototipo a escala de una herramienta para soportar confidencialidad e integridad.

Problemática:

Supondremos que una compañía de transportes ofrece a sus clientes el servicio de recogida y entrega de paquetes a domicilio. La compañía cuenta con un servidor para soportar la operación de manejo de paquetes y las consultas de los usuarios. Para el manejo de paquetes y unidades de distribución, tanto los puntos de atención al cliente como las unidades de distribución se comunican periódicamente con el servidor para informar su estado. El servidor almacena la información y atiende consultas relacionadas con estos datos vía internet.



Implementación del Prototipo:

Para este caso nos concentraremos en el proceso de atención a las consultas de los usuarios. Su tarea consiste en construir los programas servidor y cliente que reciben y responden las solicitudes de los clientes.

Servidor:

- Es un programa que guarda un identificador de cliente, un identificador de paquete y el estado de cada paquete recogido (los estados posibles son: ENOFICINA, RECOGIDO, ENCLASIFICACION, DESPACHADO, ENENTREGA, ENTREGADO, DESCONOCIDO) y responde las consultas de los clientes.
- Los estados deben representarse como constantes numéricas, excepto al presentarse mensajes a un usuario, allí deben presentarse en formato alfabético.
- Para simplificar el problema, tendremos un servidor con una tabla predefinida con: login de usuario, identificador de paquete y estado de 32 paquetes. Para consultar el estado de un paquete, un cliente envía al servidor su identificador y el identificador de un paquete, el servidor recibe los datos, consulta la información guardada y responde con el estado correspondiente. Si el identificador de usuario y paquete no corresponden a una entrada en la tabla, el servidor retorna el estado DESCONOCIDO.
- Es un servidor concurrente. El servidor principal crea los delegados por conexión al recibir cada cliente.

Cliente:

- Es un programa que envía una consulta al servidor, espera la respuesta y al recibirla la valida. Si la respuesta pasa el chequeo entonces despliega la repuesta en pantalla, si no pasa el chequeo entonces despliega el mensaje ¨Error en la consulta".
- Se sugiere manejarlo de forma concurrente.

Tanto servidor como cliente deben cumplir con las siguientes condiciones:

- Las comunicaciones entre cliente-servidor deben usar sockets y seguir el protocolo indicado.

- No use librerías especiales, solo las librerías estándar.
- Desarrolle en Java (java.security y javax.crypto).
- En el código del servidor incluya un menú que tenga dos opciones:
    - Opción 1: Generar la pareja de llaves asimétricas del servidor y almacenarlas en dos archivos.
    - Tenga en cuenta:
        - En una instalación real los permisos asignados a estos archivos serían fundamentales; la llave pública puede ser leída por cualquiera, pero la llave privada solo debería estar al alcance del propietario.
        - En este prototipo deberá copiar el archivo de la llave pública a un directorio donde los clientes la puedan leer.
    - Opción 2: Ejecutar creando los delegados, cada delegado sigue el protocolo descrito.

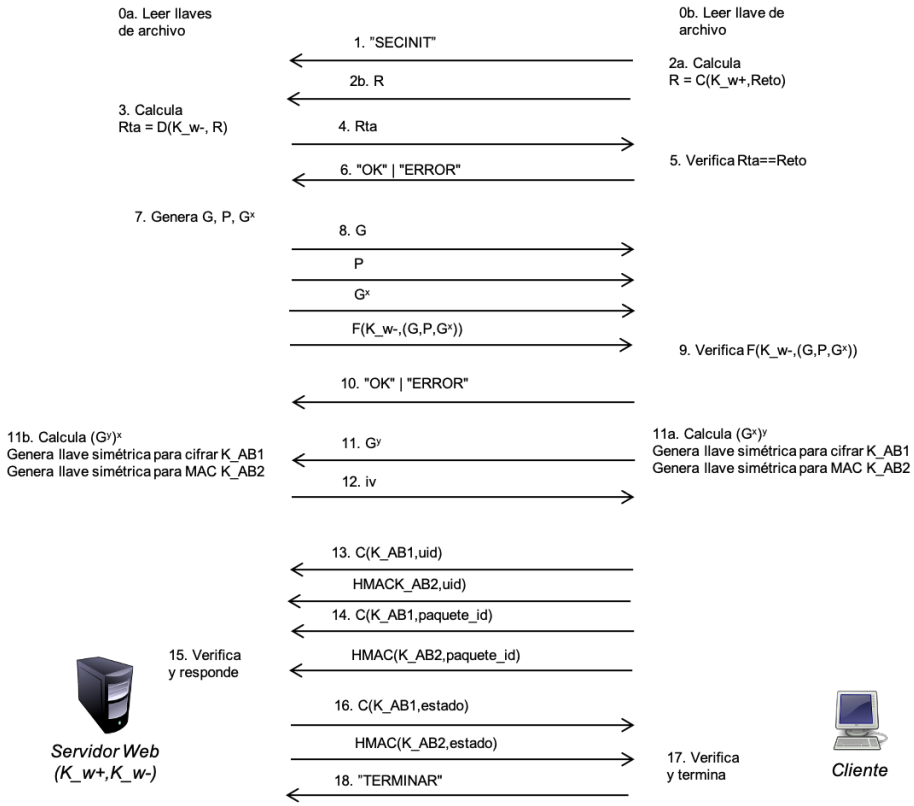


Figura 1. Protocolo de Comunicación entre Servidor y Cliente.

PARA TENER EN CUENTA:

- Algoritmos que usaremos:
    - AES. Modo CBC, esquema de relleno PKCS5, llave de 256 bits.
    - RSA. Llave de 1024.
    - El vector de inicialización para cifrado simétrico (iv) debe tener 16 bytes generados aleatoriamente
    - Firma: SHA1withRSA
    - HMACSHA384.

- Para generar las llaves simétricas:
    - Primero calcule una llave maestra por medio de Diffie-Hellman. Para obtener el valor de p y el valor del generador g, raíz primitiva de p, puede usar openssl que es un toolkit para desarrolladores de temas de seguridad. Puede encontrarla online en https://www.cryptool.org/en/cto/openssl. Para obtener los valores de p y g ejecute el comando: openssl dhparam -text 1024
    - Para manejar números grandes use la clase BigInteger de Java. Los números primos deben tener 1024 bits de longitud
    - Luego use la llave maestra para calcular un digest con SHA-512

- Parta el digest en dos mitades para generar la llave para cifrado y la llave para código de autenticación: usar los primeros 256 bits se deben usar para construir la llave para cifrar, y los últimos 256 bits para construir la llave para generar el código HMAC

Adicionalmente, resuelva las siguientes tareas:

1. Corra su programa en diferentes escenarios y mida el tiempo que el servidor requiere para: (i) responder el reto, (ii) generar G, P y G^x , (iii) verificar la consulta. Los escenarios son:
   (i) Un servidor iterativo y un cliente iterativo. El cliente genera 32 consultas.
   (ii) Un servidor y un cliente que implementen delegados. El número de delegados, tanto servidores como clientes, debe variar entre 4, 8, y 32 delegados concurrentes. Cada cliente genera una sola solicitud.
2. Construya una tabla con los datos recopilados. Tenga en cuenta que necesitará correr cada escenario en más de una ocasión para validar los resultados.
3. Mida el tiempo que el servidor requiere para cifrar el estado del paquete con cifrado simétrico y con cifrado asimétrico. Observe que el cifrado asimétrico de este valor no se usa en el protocolo, solo se calculará para posteriormente comparar los tiempos.
4. Construya las siguientes gráficas:
   (i) Una que compare los tiempos para descifrar el reto en los diferentes escenarios
   (ii) Una que compare los tiempos para generar G, P y G^x en los diferentes escenarios
   (iii) Una que muestre los tiempos para verificar la consulta en los diferentes escenarios
   (iv) Una que muestre los tiempos para el caso simétrico y el caso asimétrico en los diferentes escenarios
5. Escriba sus comentarios sobre las gráficas, explicando los comportamientos observados.
6. Identifique la velocidad de su procesador, y estime cuántas operaciones de cifrado puede realizar su máquina por segundo, en el caso evaluado de cifrado simétrico y cifrado asimétrico. Escriba todos sus cálculos.

Entrega:

- Cada grupo debe entregar un zip con:
  - Archivos fuente con la implementación del prototipo del cliente y el servidor y los archivos con las llaves del servidor (pública y privada).
  - Un subdirectorio docs con un informe que incluya: (i) la descripción de la organización de los archivos en el zip, (ii) las instrucciones para correr servidor y cliente, incluyendo cómo configurar el número de clientes concurrentes, (iii) tablas de datos, (iv) gráficas, y (v) respuestas a las preguntas planteadas.
  - **Al comienzo del informe, deben estar los nombres y carnés de los integrantes del grupo**. Si un integrante no aparece en el documento entregado, el grupo podrá informarlo posteriormente. Sin embargo, habrá una penalización: la calificación asignada será distribuida (dividida de forma equitativa) entre los integrantes del grupo.
- Recuerde incluir en su informe **todas las referencias** que use para resolver este proyecto.
- El trabajo se debe realizar en grupos de **2** personas y puede conformar los grupos de trabajo por su cuenta. No debe haber consultas entre grupos.
- Para el caso 4, también podrá conformar su propio grupo de 2 integrantes, pero el grupo del caso 4 debe ser diferente al grupo del caso 3.
- El grupo responde solidariamente por el contenido de todo el trabajo, y lo elabora conjuntamente (no es trabajo en grupo repartirse puntos o trabajos diferentes). Se puede solicitar una sustentación a cualquier miembro del grupo sobre cualquier parte del trabajo. Dicha sustentación será parte de la calificación de todos los miembros. Tenga en cuenta que desarrollar sus habilidades para trabajo en grupo le puede ayudar en su futuro profesional para integrarse más fácilmente a un grupo de trabajo y trabajar de forma productiva.
- En el parcial se incluirá una pregunta sobre el caso
- El proyecto debe ser entregado en bloqueneon.
- **La fecha límite de entrega es noviembre 5, a las 23:50 p.m.**

Referencias:

- *Cryptography and network security*, W. Stallings, Ed. Prentice Hall, 2003.
- *Computer Networks*. Andrew S. Tanenbaum. Cuarta edición. Prentice Hall 2003, Caps 7, 8.
- *Blowfish*. Página oficial es: http://www.schneier.com/blowfish.html
- *RSA.* Puede encontrar más información en: http://www.rsa.com/rsalabs/node.asp?id=2125
- *CD X509*. Puede encontrar la especificación en: http://tools.ietf.org/rfc/rfc5280.txt
- *MD5*. Puede encontrar la especificación en : http://www.ietf.org/rfc/rfc1321.txt