Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional
Semestre 2024-10

Caso de Estudio 3 – Canales Seguros

Objetivos

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

Problemática:

Suponga que un servicio soporta consultas en línea para sus usuarios, garantizando confidencialidad e integridad. Su tarea consiste en construir un cliente y un servidor que corren de acuerdo con el protocolo descrito más adelante (figuras 1 y 2).

- El servidor es un proceso que recibe y responde las consultas de los clientes. Para simplificar esta parte del problema y concentrarnos en los aspectos de seguridad, la consulta solo será un número y el servidor debe responder con número-1.
- El cliente es un proceso que envía una consulta al servidor, espera la respuesta y la presenta al usuario.

Tanto servidor como cliente deben cumplir con las siguientes condiciones:

- Las comunicaciones entre cliente-servidor deben usar sockets y deben estar cifradas de acuerdo con el protocolo definido.
- Desarrolle en Java. No use librerías especiales, solo las librerías estándar (java.security, javax.crypto).
- Generaremos por adelantado las llaves pública y privada del servidor. Tenga en cuenta que en una instalación real la llave pública puede ser leída por cualquiera, pero la llave privada solo debería estar al alcance del propietario.

PARA TENER EN CUENTA:

- Como algoritmos usaremos:
  - Cifrar - C(..) en las figuras 1 y 2: AES. Modo CBC, esquema de relleno PKCS5Padding, llave de 256 bits.
  - Firma - F(..) en las figuras 1 y 2. SHA256withRSA
  - Código de autenticación - HMAC(..) en la figura 1: HMACSHA256

- Para generar las llaves necesarias:
  - 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
  - Si necesita 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
  - El vector de inicialización para cifrado simétrico (iv) debe tener 16 bytes generados aleatoriamente

El cliente y el servidor deben crear delegados y los delegados deben comunicarse siguiendo el protocolo descrito en las figuras 1 y 2. El servidor principal crea los delegados por conexión al recibir cada cliente.

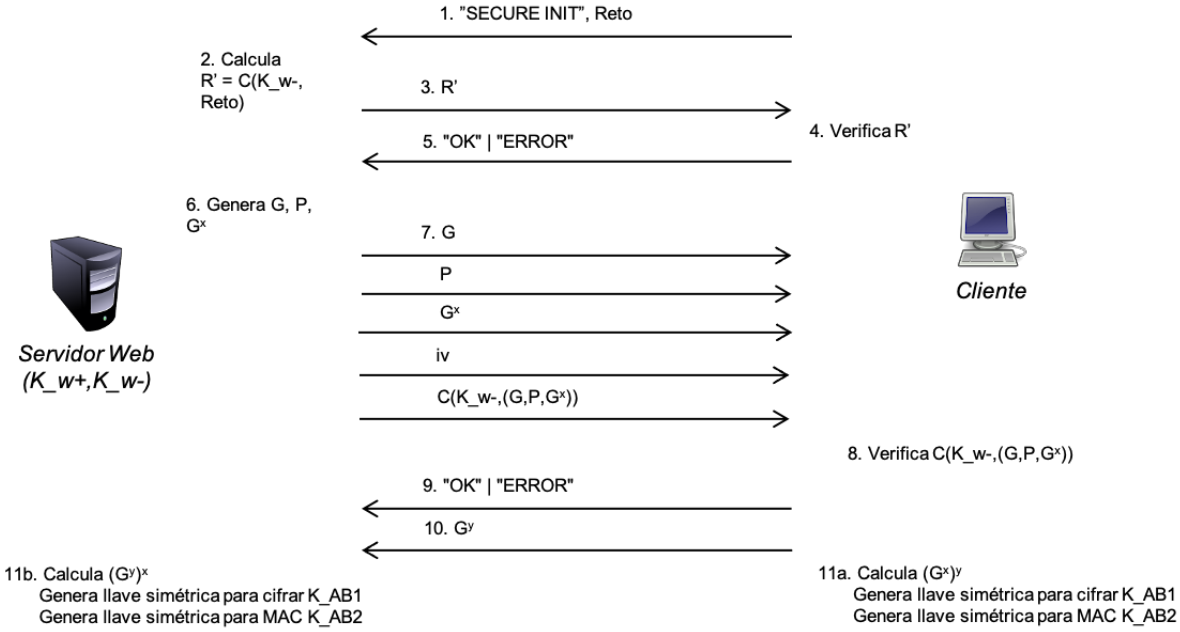

Figura 1. Protocolo de comunicación entre el servidor y el cliente.

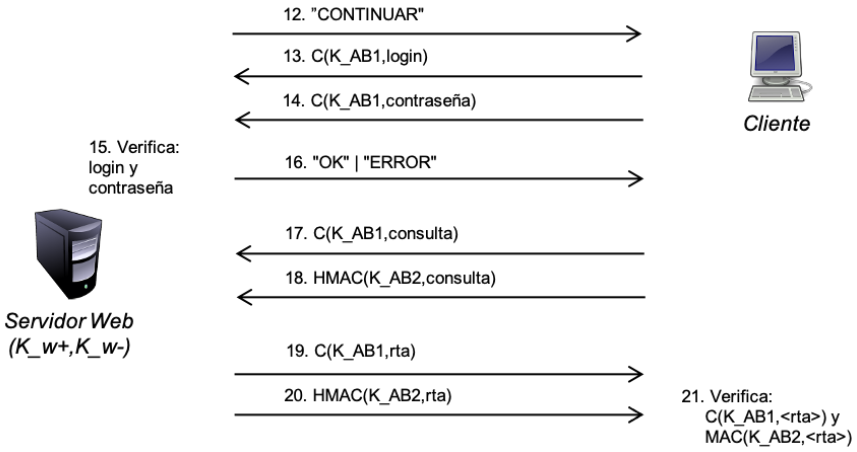

Figura 2. Protocolo de comunicación entre el servidor y el cliente (parte 2).

Si el cliente encuentra un error en la validación debe enviar el mensaje “ERROR” al servidor y terminar (p.e. ver pasos 5 y 9 en la figura). Así mismo, si el servidor encuentra un error en la validación enviará el mensaje “ERROR” al cliente y terminará la conexión (paso 16).

Responda las siguientes preguntas:

1. Para responder las siguientes preguntas puede usar fuentes externas, solo recuerde construir las respuestas con sus propias palabras y citar las fuentes usadas:
   - (i) 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?
   - (ii) ¿Por qué es necesario cifrar G y P con la llave privada?
   - (iii) 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).

2. Corra su programa en los diferentes escenarios y mida el tiempo que el cliente demora para:
   (i) Verificar la firma
   (ii) Calcular G^y
   (iii) Cifrar la consulta
   (iv) Generar el código de autenticación
   (los escenarios deben variar el número de clientes delegados debe variar entre 4, 16 y 32 al menos).
3. En los mismos escenarios considerados en el punto anterior mida el tiempo que el servidor demora para:
   (i) Generar la firma
   (ii) Descifrar la consulta
   (iii) Verificar el código de autenticación
4. Construya una tabla con los datos recopilados (tenga en cuenta que necesitará correr cada escenario en más de una ocasión para validar los resultados) y luego construya una gráfica con los datos de la tabla.
5. Escriba sus comentarios sobre los resultados.
6. Identifique la velocidad de su procesador, y estime cuántas consultas puede cifrar su máquina, cuántos códigos de autenticación puede calcular y cuántas verificaciones de firma, por segundo. Escriba todos sus cálculos.

Entrega:

- Cada grupo debe entregar un zip con:
  - Los archivos fuente con la implementación del prototipo del cliente y el servidor.
  - 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 el servidor y el cliente, incluyendo cómo configurar el número de clientes concurrentes (iii) las respuestas a las preguntas.
  - **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 **3** personas.
- 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.
- Para los casos 3 y 4 del curso está permitido conformar los grupos de trabajo por su cuenta. Pero, al menos un integrante debe cambiar en la conformación de los grupos para los dos casos.
  - Si un grupo no cumple esta regla, entonces la calificación del caso 4 se multiplicará por 0,8 (20% de penalización)
- En el parcial se incluirá una pregunta sobre el desarrollo de alguna de las funcionalidades del caso
- El proyecto debe ser entregado en bloqueneon.
- **La fecha límite de entrega es abril 30, 2024 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