**Caso de Estudio 3 – Canales Seguros**

**Sistema de rastreo de paquetes en una compañía transportadora**

1. **Organización de los archivos en el archivo .zip**

El archivo **.zip** contiene las siguientes carpetas,

* **Cliente:** Carpeta con la implementación del Cliente. Contiene la carpeta **data** que contiene la *llave pública asimétrica*. La carpeta **src,** almacena los siguientes archivos.
  + **Cliente.java,** que ejecuta el Cliente
  + **ProtocoloCliente.java,** que es llamado por **Client.java** para iniciar su protocolo**.**
* **Servidor:** Carpeta con la implementación del Servidor. Contiene las carpetas **data** que contiene las *llaves asimétricas pública y privada*. La carpeta **src,** almacena los siguientes archivos,
  + **Servidor.java,** que ejecuta el Servidor.
  + **ThreadServidor.java,** clase querepresenta un hilo que corre el Servidor.
  + **ProtocoloServidor.java,** que es llamado por **Servidor.java** para iniciar su protocolo**.**
  + **Paquetes.java,** clase que define los paquetes a enviar en el protocolo.

1. **Instrucciones para correr el prototipo**

A continuación, se presentan las instrucciones para la ejecución del prototipo Cliente/Servidor.

1. Ejecutar el archivo **Servidor.java** ubicado en la carpeta **Servidor;** Se crearán los archivos de las llaves *Pública* y *Privada* y se almacenarán en la carpeta **data.**
2. Copiar el archivo **publicK.txt** ubicado en la carpeta **data**, de **Servidor.**
3. Ejecutarel archivo **Cliente.java** ubicado en la carpeta **Cliente** y enviar la petición de inicio, “INICIO”.
4. Esperar el mensaje de confirmación y enviar el reto al *servidor.*
5. **Esquema para generar llaves**

En el desarrollo del prototipo, se siguió un esquema para generar las llaves simétricas y otro para generar llaves asimétricas.

**Llave simétrica**

* Se utiliza el algoritmo **AES** para generar la llave simétrica.
* El proceso se ejecuta en la clase **ProtocoloCliente.java** del Cliente.

**Llaves asimétricas**

* Se utiliza el algoritmo **RSA** con un tamaño de llave 1024para generar las llaves asimétricas.
* Se almacenan las llaves en un archivo *.txt* y se almacenan en la carpeta **data** del Servidor; **privateK.txt** almacena la *llave privada* y **publicK.txt** almacena la *llave pública*.
* Si el Cliente lo necesita, recupera el archivo *.txt* de la llave pública y lo lee hasta obtener la llave.

1. **Conclusiones y resultados**

Tras completar el reto, observamos los siguientes resultados por etapa:

* **ETAPA 1:**

En esta etapa se generó con éxito las **llaves asimétricas**. Fue posible **cifrar el reto** enviado por el Cliente, y al descifrarlo obtener los mismos valores sin que se presente perdida de paquetes.

* **ETAPA 2:**

El Cliente generó **la llave simétrica** y la **cifró**, exitosamente, con la llave pública generada por el Servidor. Lamentablemente, al momento de enviar la llave cifrada al Servidor, es posible que se presente pérdida de paquetes y por esta razón, no se descifra la llave correctamente.

* **ETAPA 3:**

En la etapa de consultas, el Cliente logra **enviar su nombre** para consultarlo en la tabla, esperando **recibir** el mensaje de confirmación **“ACK”**. Seguido, el Cliente nuevamente intenta **enviar el id del paquete**, el servidor consulta el nombre de usuario y el identificador del paquete en la tabla y se esperaba que este responda exitosamente.

Finalmente, se codificó la etapa para que envíe el código de resumen o digest, pero no fue posible probarlo durante la ejecución del prototipo.