Objetivos	Programación de sockets TCP utilizando encriptación para trasmitir información.

Índice

---

1. Introducción

En esta práctica se van a conectar dos máquinas utilizando el protocolo TCP. Se va a utilizar un proveedor de clases de algoritmos de seguridad para encriptar la información que se envía desde el cliente al servidor.

1.1. Criptografía de clave pública

Para utilizar la criptografía de clave pública se necesita generar un par de claves: una clave pública y otra privada.

Para cifrar datos se utiliza la clave pública del destinatario de los datos. Sólo el destinatario (que dispondrá de la clave privada correspondiente) podrá descifrar los datos.

La criptografía de clave pública tiene los siguientes problemas:

- Los algoritmos de cifrado mediante clave pública son generalmente más lentos que los de clave simétrica y requieren una clave muy larga y una forma de generar números primos grandes para generar la clave, haciendo que sean más costosos computacionalmente hablando.
- La clave privada debe ser almacenada de forma segura ya que si cae en manos de una tercera persona, esta podrá descifrar y firmar mensajes suplantando a quién generó las claves.

- Es difícil autenticar el origen de una clave pública. Esto se puede solucionar con los certificados.

Algoritmos como RSA, Diffie-Hellman y El-Gamal implementan la metodología de cifrado mediante clave pública.

1.2. Criptografía de clave simétrica

La criptografía de clave simétrica (la misma clave para cifrar y para descifrar) es generalmente más rápida y simple de implementar. Inicialmente se acuerda entre las dos partes cual será la clave y el algoritmo a utilizar. Posteriormente se puede cifrar o descifrar datos utilizando la clave.

La desventaja del cifrado mediante clave simétrica estriba en la dificultad de enviar la clave simétrica mediante un canal seguro.

Implementaciones comunes de los algoritmos de clave simétrica son DES (Data Encryption Standard), 3-DES (triple DES), IDEA, RC5 Blowfish y AES (Advanced Encryption Standard).

1.3. Combinando ambas

En la práctica lo habitual es combinar ambas formas de cifrado. El algoritmo de cifrado mediante clave pública es utilizado para cifrar la clave simétrica. Esta clave simétrica cifrada puede ser enviada a la otra parte para que con su clave privada la descifre. Una vez que las dos partes tienen la clave pueden compartir la información cifrada utilizando un algoritmo de clave simétrica como 3-DES, AES, IDEA o Blowfish, véase la figura 1.

2. Tareas

Se trata de realizar una clase `ServidorSeguro.java` que sea un servidor para que se comunique con un cliente que se proporciona. El protocolo a utilizar para la comunicación será TCP. La información sensible se enviará codificada utilizando un algoritmo de cifrado de clave simétrica.

El cliente realiza las siguientes tareas:

- Se conecta al servidor.
- Genera una clave simétrica para ser utilizada con el algoritmo AES.
- Envuelve (cifra) esta clave con la clave pública del servidor utilizando RSA.
- Envía al servidor la longitud de la clave cifrada.
- Envía al servidor la clave cifrada.

- Presenta al usuario una GUI con un área de texto y dos botones.
- Cuando el usuario pulsa el botón "Enviar"se cifra el texto con la clave simétrica utilizando el algoritmo AES. Se envía al servidor un texto para indicar inicio de trasmisión de mensaje, se envía la longitud de texto cifrado y se envía el texto cifrado.

El servidor a implementar deberá realizar las siguientes tareas:

- Acepta la conexión del cliente.
- Lee la longitud de la clave cifrada.
- Lee en un vector de bytes (de la longitud leída anteriormente) la clave simétrica cifrada).
- Utiliza su clave pública para descifrar la clave secreta utilizando el algoritmo RSA.
- Lanza un hilo para tratar la lectura (puesto que toda la información no se recibe de una vez sino que se recibe conforme el usuario va pulsando el botón "Enviar").
- En ese hilo el método `run()` comprueba si se ha recibido la marca de inicio de trasmisión de mensaje. Si es así lee la longitud del mensaje cifrado, lee en un vector de bytes (de la longitud leída anteriormente), descifra utilizando el algoritmo AES el mensaje y lo muestra por pantalla.

El proceso está ilustrado en la figura 2

Los ficheros proporcionados son:

- ClienteSeguro.java Esta clase es el cliente que se conecta vía TCP al servidor que se debe implementar.
- Se puede utilizar la clase `GeneraClavesRSA.java` para generar el par de claves. Esta clase se proporcionó con los ejemplos de encriptación en el tema sobre Applets.

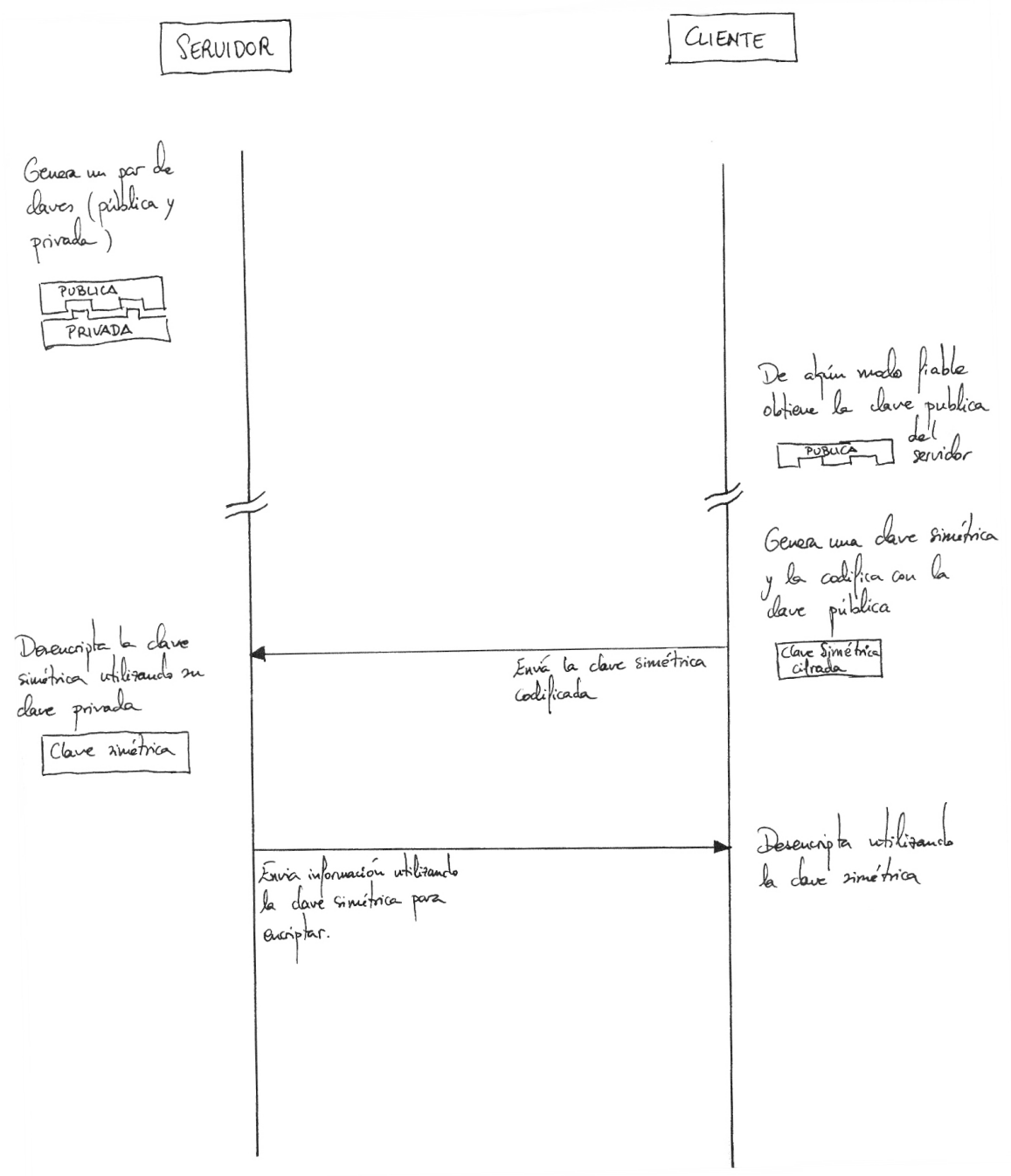

Figura 1: Descripción general de la utilización de cifrado utilizando clave pública y simétrica.

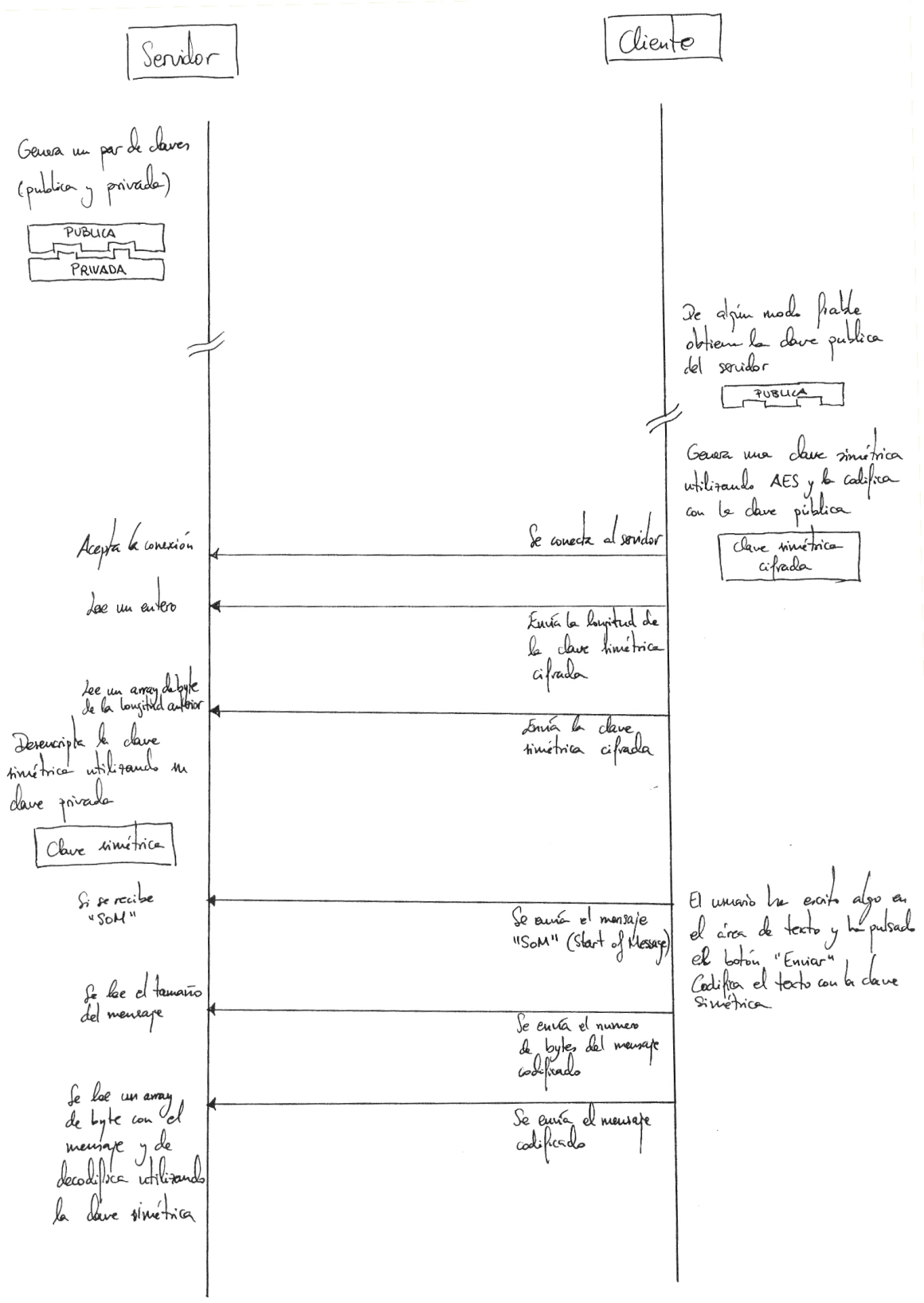

Figura 2: Descripción detallada del cliente proporcionado y de lo que debe realizar el servidor a implementar.