Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Redes de Computadoras I (CI-4835)
Trimestre Enero-Marzo 2016

MINIPROYECTO N° 1

Objetivo General: Proveer al estudiante de un trabajo práctico de diseño y construcción de una aplicación final, que use el paradigma de operación Cliente – Servidor, para que este comprenda, en general, el funcionamiento de aplicaciones y servicios en redes.

Objetivos Específicos: Al finalizar esta actividad el estudiante deberá estar en capacidad de:

• Comprender el uso y la programación de la Interfaz de Aplicaciones (API) Sockets de Berkeley.
• Entender el uso y programación de las llamadas a Procedimientos Remotos (RPC).

Parte I.-

Introducción

De acuerdo a Bruce Schneier, autor de los libros “Criptografía Aplicada” y “Secretos y Mentiras en un mundo digital de redes”, la criptografía es el arte y la ciencia de mantener mensajes seguros. En los postulados iniciales de esta ciencia, un algoritmo criptográfico, también llamado cifrador, es una función matemática usada para encripción y decripción de un mensaje. Sin embargo este esquema tiene la desventaja de que si algún usuario (espía) estudia el algoritmo de cifrado y logra entender su funcionamiento, a partir de ese momento termina la seguridad del algoritmo. Por ello, la criptografía moderna resuelve este problema incorporando una clave de usuario, donde tanto la operación de cifrado como la de descifrado, usan esa misma clave para realizar sus operaciones. Lo que debe ser secreto es entonces la clave y no el cifrador, se da lugar así a lo que se conoce como un algoritmo simétrico, algoritmo de clave secreta o algoritmo de una sola clave. La Figura 1.muestra una representación de este algoritmo.

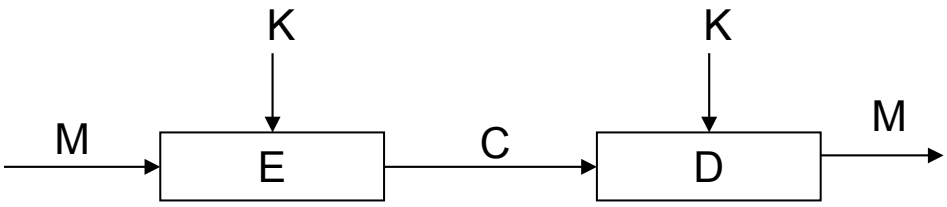

Figura 1. Encripción y Decripción con Clave

Donde:

E= Función de Encripción
K= Clave de cifrado
M=Mensaje en texto plano
C= Texto cifrado
D = Función de Decripción

Utilizando una notación matemática se puede expresar de la siguiente manera:

$$E_K(M)=C$$

$$D_K(C)=M$$

Por su parte, un algoritmo conocido de clave simétrica que funciona desde la época del emperador romano Julio César, es el famoso cifrado de César, que consiste en que cada carácter de un texto plano es reemplazado a partir del tercer caracter del alfabeto en dirección a la derecha. Es decir cualquier carácter *A* es sustituido por *D*, cualquier *B* es sustituido por *E*, . . . , cualquier *W* es sustituido por *Z*, cualquier *X* es sustituido por *A*, cualquier *Y* es sustituido por *B* y cualquier *Z* es sustituido por *C*. La Figura 2 ilustra el funcionamiento de este cifrado.

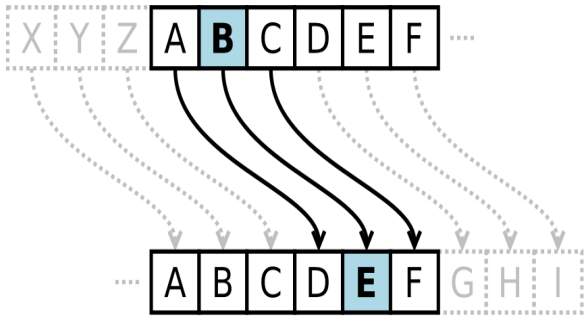

Figura 2. Cifrado de César con desplazamiento de 3 espacios

Para este caso, el cifrador puede ser conocido, pero lo que debe mantenerse en secreto, es decir la clave, es el valor del desplazamiento del alfabeto. En la Figura 2, se trata del valor "3".

Descripción del trabajo a desarrollar

Se desea que usted desarrolle una aplicación de usuario sencilla usando Sockets, en lenguaje C, tal que permita instrumentar un *Sistema de cifrado y descifrado de archivos de textos* (SCDAX). El mismo deberá operar de acuerdo a una variación de la descripción, dada anteriormente sobre el funcionamiento del cifrado de César. Se espera que se implemente un programa servidor que deberá operar en modo *concurrente* a fin de poder dar atención a varios programas clientes que requieran el servicio, tal como se muestra en la Figura 3.

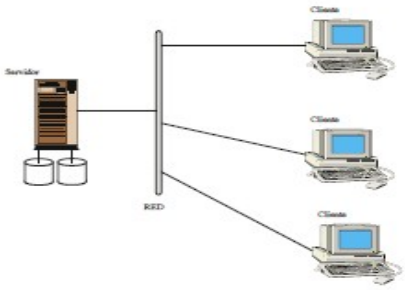

Figura 3. Clientes que se comunican con el servidor a fin de requerir el servicio de cifrado/descifrado.

La idea principal es que los usuarios dispongan de un servicio de apoyo para almacenar localmente, en modo confiable, un documento. El servidor proveerá la información que se le suministre en modo cifrado o la descifrará en caso de que eso le sea solicitado. En otras palabras, la confidencialidad de la información será necesaria únicamente para el almacenamiento, no para la transmisión y recepción de la información. Es decir, se asume que los canales de comunicación son seguros[1].

Funcionamiento del Sistema

El sistema de cifrado y descifrado SCDAX estará implementado sobre una red de computadores, donde tanto los clientes como el servidor podrán ejecutarse en máquinas diferentes. Un usuario conectado a una computadora que requiera cifrar una información almacenada en un archivo, por lo tanto ejecutará el comando del cliente para llevar a cabo esta tarea. Si al servidor le es solicitado cifrar una información, esta debe estar sólo en formato de texto, en caso contrario, deberá enviar un mensaje al cliente indicando que no podrá realizar el cifrado debido a que no es un archivo de texto.

De modo que el servidor tiene dos funciones: cifrar y descifrar información y sólo podrá descifrar una información que haya sido cifrada por el mismo, de manera que el servidor debe utilizar un mecanismo (esteganografía[2]) de

1 En la realidad eso no es así. Se asume únicamente con el propósito de simplificar el trabajo y facilitar la actividad pedagógica.

2 Esteganografía: Trata el estudio y aplicación de técnicas que permiten ocultar mensajes u objetos, dentro de otros, llamados ***portadores,*** de modo que no se perciba su existencia. Es decir, procura ocultar mensajes dentro de otros objetos y de esta forma establecer un canal encubierto de comunicación, de modo que el propio acto de la comunicación pase **inadvertido** para observadores que tienen acceso a ese canal. Fuente: https://es.wikipedia.org/wiki/Esteganograf%C3%ADa

reconocimiento de cifrado que le permita identificar su procedimiento. Cuando al servidor le sea solicitado cifrar una información, le devolverá al cliente el texto cifrado y si el requerimiento es descifrar un texto cifrado, retornará al cliente el texto plano correspondiente. El cliente debe mostrar la información por pantalla y además almacenarla en un archivo.

El procedimiento que deberá seguir el servidor para realizar el cifrado o descifrado considerará que el alfabeto a usar contiene 27 letras, luego el servidor debe conocer la longitud de la clave y en que dirección deberá realizar la sustitución, a la izquierda o a la derecha. Ejemplo:

El texto a cifrar es: HOLA MUNDO, la clave es 5 y la dirección es a la derecha

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U
U	T	S	R	Q	P	O	Ñ	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V

Luego el texto cifrado a la derecha quedaría: CKGV HPIYK y a la izquierda: ÑGKU JAIRG.

Finalmente es muy importante tener en cuenta que el servidor debe registrar todas las actividades, cada vez que un cliente intenta o logra conectarse a el, así como también cualquier problema o error que haya ocurrido durante la conexión. Por lo tanto se requiere de la creación y actualización de un archivo llamado bitácora que contenga la siguiente información:

Por cada entrada del archivo bitácora, el primer elemento indicará la fecha y hora del mensaje, el segundo elemento la actividad ejecutada, el tercer elemento corresponderá a la dirección IP del cliente y el último elemento reportará el mensaje. Un ejemplo de esto sería:

[Lun Ene 11 15:32:25 2016] [error] [cliente 159.90.34.13] Acceso negado por el servidor.

Normas para el trabajo

Cada grupo debe crear su propio código de mensaje y ello deberá quedar claramente establecido en la documentación que será entregada en papel. Adicionalmente, la aplicación deberá emplear un conjunto de comandos específicos, cuya sintáxis y operación se describe seguidamente al igual que un protocolo para su funcionamiento en entidades distintas.

Sintaxis de la invocación de Comandos:

La sintaxis para la invocación de la ejecución del servicio que ofrece SCDAX deberá ajustarse a las siguientes indicaciones:

scdax_svr -l <puerto_scdax_svr> -b <archivo_bitácora>
Donde:
<**puerto_scdax_svr**> Es el número de puerto local por donde el servidor atenderá la interacción.

<**archivo_bitácora**> Es el nombre y dirección relativa o absoluta de un archivo de texto que realiza operaciones de bitácora. Cada línea del archivo, es una entrada que incluye como elementos mínimos la fecha, la hora y el evento a reportar.

La sintaxis del módulo que ejecutarán los usuarios finales:

scdax_cli -i <dir_ip> -p <puerto_scdax_svr> -c <long_clave> -a <dir_cif> -f <archivo_a_procesar>

Donde:
<dir_ip>: Es el nombre de dominio o la dirección IP (versión 4) del equipo en donde se deberá ejecutar el servidor.

<puerto_scdax_svr>: Es el número de puerto remoto a través del cual el servidor atenderá la comunicación solicitada.

<long_clave>: Corresponde al valor de la clave de cifrado y es un valor entero entre 1 y 27.

<dir_cif>: Es la dirección del cifrado, puede tomar sólo 2 valores, derecha o izquierda.

<archivo_a_procesar>: Es el archivo que contiene la información a ser cifrada o descifrada.

Diseño del protocolo de comunicación:

Para la definición del protocolo se deben tener en cuenta los siguientes aspectos y como parte de la documentación se deberá describir en forma clara, cada uno de ellos:

1. ¿Qué tipo de sockets decidió emplear?. Justifique su respuesta.

2. Identifique todos los mensajes del sistema, indicando: el formato del mismo, su tamaño en bytes, quién genera el mensaje y quién lo recibe y procesa.

3. Indique la estructura en pseudocódigo que tendrían los procesos que se ejecutan en los diferentes componentes del sistema, indicando las llamadas de la biblioteca de sockets que hay que utilizar en los distintos elementos de su aplicación.

4. Realice el diseño completo del protocolo de comunicación que construya e ilustre como opera el mismo. Incluya el punto de vista de los cambios de

estado de las entidades que se comunican. Se sugiere usar grafos dirigidos para facilitar la explicación de como es la actividad del protocolo.

5. Describa aspectos del proyecto que funcionan según el enunciado y cuales no. Cualquier requerimiento no desarrollado o que contenga fallas, deberá ser señalado claramente.

Esta sección debe imprimirse y será entregada el viernes de semana 8 a la profesora del laboratorio, o en su defecto, puede dejarlo en el casillero ubicado en el departamento de computación y TI, de 8:30 a.m. a 3:45 p.m. Sea organizado en el documento que será entregado, incluya portada, contenido, descripción del problema a resolver, referencias bibliográficas, entre otros; se evaluará la presentación y estructura.

Consideraciones Adicionales:

* Durante el funcionamiento del sistema puede ser que el servidor no esté siempre operativo. En ese caso el cliente hará la petición hasta 3 veces, si el servidor no le responde, entonces enviará un mensaje a la pantalla indicando que el tiempo de repuesta se agotó.

* El servidor deberá operar en forma continua.

* Todos los parámetros estarán siempre presentes, pero pueden estar presentes en la invocación en un orden diferente al que se indicó en la sintáxis. Aún así, el sistema deberá operar correctamente**.**

Plataforma donde será evaluado su trabajo

Los proyectos serán evaluados en los equipos del Laboratorio de Computación (LDC). Las máquinas que se encuentran operativas para hacer las pruebas bajo el dominio ldc.usb.ve son: auliya, tuek, harq, irulan y varota. Aquellos estudiantes que no posean cuentas en el laboratorio de computación deben solicitar la creación de su cuenta.

Para las pruebas de su proyecto utilice como número de puertos lógicos, un número de 5 cifras que comience por 2 y donde las siguientes 4 cifras coincidan con las últimas cuatro cifras del número de carné de alguno de los integrantes del equipo. Si se requiere mas de un puerto, repita el procedimiento pero con el numero de carné de otro integrante.

La entrega de la aplicación debe hacerse de manera electrónica a través del Aula Virtual, en la fecha y hora indicada: viernes de semana 8 antes de las 10:30 p.m. El acceso para subir los mini proyectos será cerrado luego de la hora establecida, por lo tanto se recomienda no esperar hasta la última hora para subir los archivos.

La entrega será con un archivo comprimido que se identifique con nombre Apellido1_Apellido2_X.tar.gz

(ver http://ldc.usb.ve/~figueira/Cursos/ci3825/taller/material/TGZ.html de cómo hacer un archivo tar.gz), donde X será reemplazado por los términos Sockets o RPC y Apellido1 y Apellido2 corresponden a los apellidos de los integrantes del grupo (por ejemplo, Mora_Azuaje_Sockets.tar.gz) con los siguientes archivos:

•Archivos fuente. Por ejemplo, si se trata de un proyecto en lenguaje C, debe incluir todos los ".c" y ".h.
•Makefile
•Un archivo LEEME.txt, que explique el contenido del "tar.gz":

Tenga cuidado de no incluir ejecutables, archivos objeto (.o) o cualquier otro archivo generado automáticamente a partir de los fuentes.

Para efectos de desarrollo puede utilizarse una máquina para correr todos los componentes (clientes y servidores). Una vez funcione en una máquina, debe probarlo en al menos dos máquina para verificar que funcione bien de forma remota. La corrección se hará usando varias máquinas.

Código:

El código deberá incluir un documento del tipo “Leeme.txt” con al menos los siguientes elementos:
• nombres y apellidos de los integrantes del grupo
• números de carnet
• qué archivos lo componen y qué tiene cada archivo
• qué hace el programa
• cómo se ejecuta
• que condiciones particulares tiene, por ejemplo la entrada XX tiene máximo 50 caracteres" o "el máximo número de registros YY es 20"

Por su parte, el código deberá estar debidamente documentado, siguiendo los estándares de documentación de Javadoc con el programa y los encabezados de las funciones implementadas. Adicionalmente, todos sus programas deben seguir las buenas prácticas de estilo de programación en C y todas las llamadas al sistema deben ser correctamente manejadas.

Condiciones de la entrega:

* La entrega se deberá hacer dentro del lapso indicado. Cualquier demora podrá dar la potestad al docente de no aceptar el trabajo. Los informes deberán ser impresos además y se harán llegar según indique el profesor de turno. Cualquier falla de versión inapropiada, falta de algún elemento u otra situación anómala, podrá dar potestad al docente de colocar cero (0) como calificación. Recuerde, el docente no esta obligado a ajustar o adaptar su sistema. Usted debe preparar todo para facilitar la ejecución de su proyecto sin esfuerzos. No piense que tendrá otra oportunidad para arreglar cualquier falla.

***** Los equipos deben ser de dos (2) estudiantes, pero todos sus integrantes deberán asegurarse de poder explicar, a cabalidad, la distribución del trabajo de cada miembro integrante. Además, cada miembro del equipo debe estar en capacidad de comprender y desarrollar cualquier parte del trabajo, incluso aunque no le haya sido asignada originalmente. Si estas condiciones no se cumplen, el evaluador del proyecto podrá reprobar al integrante que no cumpla con las mismas.

* Cualquier caso de plagio será severamente castigado, no será evaluado ninguno de los proyectos involucrados y serán aplicadas las sanciones correspondientes establecidas en los reglamentos de la universidad.

Parte II.-

Realice los cambios pertinentes para implementar la solución del mismo problema a través de llamadas a procedimientos remotos RPC. Debe entregar también la documentación con los cambios que correspondan y el archivo comprimido.

Finalmente, es posible que el equipo evaluador distribuya otro documento que complemente la información o detalle cualquier aspecto que sea necesario aclarar. Por ejemplo, casos de pruebas.

Grupo Docente de Redes de Computadores I (GDRC I)
Ene-Mar 2016