Métodos simples de encriptación para conexiones con Socket

Cargnelutti, Pablo Rubén
Universidad tecnológica Nacional, Facultad Regional Córdoba

Abstract:
En toda conexión mediante Sockets existe el riesgo de que personas ajenas a la comunicación, mas comúnmente llamados intrusos, puedan estar escuchando los mensajes transmitidos. En este articulo se detallaran algunos métodos simples para codificar o encriptar los mensajes transmitidos de manera que impidan su entendimiento a terceros indeseados. Los métodos explicados a continuación presentan diferentes formas de utilización durante la conexión, se detallaran también las posibles falencias de algunos métodos y posibles utilizaciones de cada uno de ellos. Los mecanismos de protección de los mensajes podrá ser aplicado a gran cantidad de aplicaciones basadas en conexiones con socket, o a mensajes particulares que requieran ser enviados por medios no seguros. También se presenta un ejemplo de una aplicación que utiliza estos métodos para comprender mejor su forma de uso en las conexiones mediante Socket tipo cliente-servidor.

Palabras Clave:
Encriptación, codificación, conexiones TCP/IP, Socket.

Introducción:

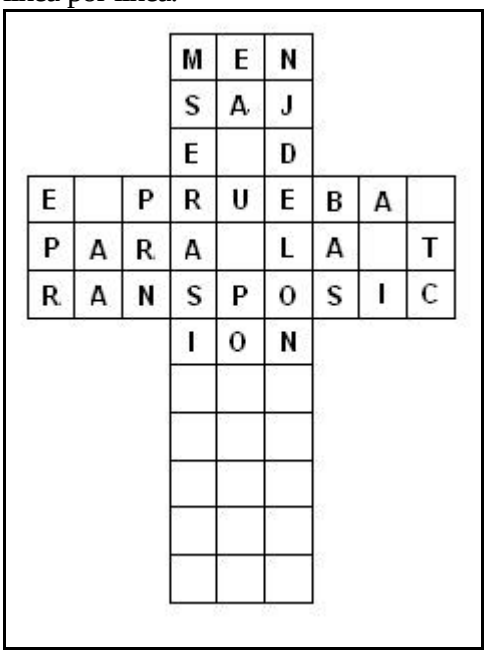
En los tiempos modernos siempre es posible que haya gente escuchando nuestras conversaciones. Posibles intrusos que interceptan los mensajes de las conexiones de nuestras PCs con el objetivo de robar datos de importancia o hacerse pasar por alguien más para tener acceso a cualquier sistema de manera remota. Para impedir que personas ajenas puedan conocer nuestros datos o leer nuestras conversaciones se crearon métodos para ocultar nuestros mensajes y hacerlos indescifrables. Existen mecanismos para autentificar a los que quieren ingresar al sistema para saber si están autorizados o no a entrar. A continuación se detallaran algunos de los métodos mas simples que cualquier persona puede implementar para ocultar su información a terceros. Desde la simple técnica de la transposición a sustituciones mas complejas. Cabe mencionar que estos métodos son para mensajes o información de baja o mediana importancia. Para proteger información muy critica es necesario utilizar métodos industriales.

Metodologías de Encriptación:

Transposición:

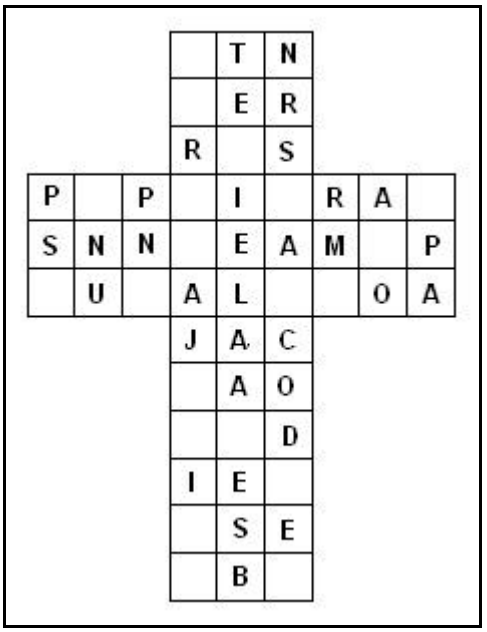
La transposición es simplemente alternar de lugar las letras de un mensaje. Generalmente no se utiliza por si sola, sino acompañada de algún método de sustitución, métodos que veremos mas adelante.
La transposición se puede utilizar en cualquier parte de la encriptación para dificultar aún más la posible lectura del mensaje o texto plano. Para realizar esto es necesaria de una clave privada, de tal manera que pueda volverse al mensaje original usando la clave de manera inversa. El método de transposición que explicare a continuación se basa en algo muy conocido por todos como el cubo de Rubik o comúnmente llamado cubo mágico.
Para realizar la transposición se utiliza un algoritmo que se detalla mas adelante. El principio de funcionamiento básico es el siguiente: tomando en cuenta el cubo, se extiende en una superficie plana. El algoritmo toma el mensaje y ubica una letra en cada cuadrado, dejando vacías las demás, si el mensaje fuera corto (ver imagen 1). Si el mensaje es mas largo se utiliza otro cubo para completarlo.
Luego de ingresado el mensaje, el algoritmo procede a su transposición según los movimientos permitidos en el cubo de Rubik, con movimientos de lados. El lado que moverá y la dirección hacia donde la moverá es totalmente aleatorio. Sin embargo la combinación de movimientos es guardada por el algoritmo para poder volver a transponer los caracteres en el destino del mensaje (ver imagen 2). El mensaje transpuesto se lee de

acuerdo a como se cargo el texto plano, o sea línea por línea.



(Imagen 1: Mensaje de prueba para la transposición)

La cantidad de movimientos que realiza el algoritmo para mezclar las letras lo puede ingresar el usuario. Del tamaño del numero dependerá lo mezclado que este el mensaje y también del tamaño de la clave que generara

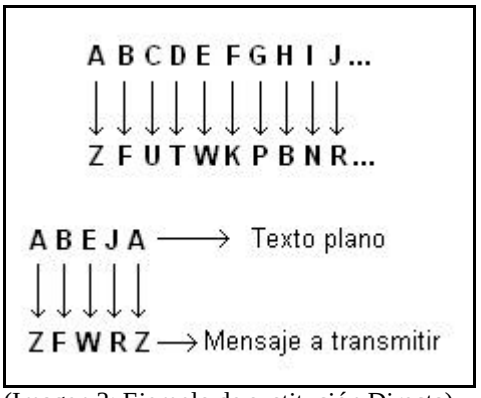


(Imagen 2: Mensaje mezclado por el algoritmo)

el algoritmo para realizar el proceso inverso y obtener el mensaje original. Este método es conveniente utilizarlo con algún método de sustitución para dificultar mas su descifrado si cayera en manos equivocadas.
Para aumentar la transposición se puede usar muchas veces este método sobre el texto plano abarcando diferentes sectores en el texto, de forma preestablecida, para que no sea imposible después su descifrado.

Sustitución directa:

Este método es el mas simple de sustitución y se basa en un principio de que cada letra tiene una correspondencia con otra. En el momento de encriptar el mensaje se sustituye la letra original por su correspondencia. Estas correspondencias se encuentran en la clave privada y si esta clave cae en diferentes manos, el mensaje puede ser leído aplicando el mismo método (ver imagen 3).



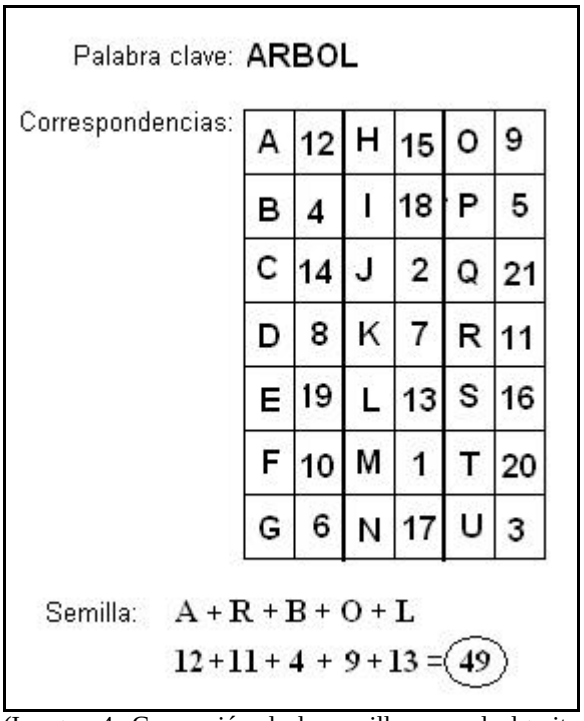
(Imagen 3: Ejemplo de sustitución Directa)

Las falencias de este método son que puede quebrarse la protección mediante el uso de técnicas probabilísticas. Estas técnicas se fijan en los caracteres que mas se repiten en un texto, asignándoles probabilidades de ocurrencia y finalmente descifrando el mensaje. Se pueden utilizar combinaciones de estos métodos para lograr formas mas seguras de proteger la información.

Encriptación por número de palabra:

En este método asignamos a cada carácter del alfabeto un valor numérico único. Estos números deben ser consecutivos. Esta asignación de números a cada letra conformara la clave que debemos ocultar. Este algoritmo lo que hace es tomar una palabra

ingresada por el usuario y sumar sus caracteres de acuerdo a sus correspondientes valores en la clave. Es resultado de esto es un único numero que se utilizara como semilla en cualquier algoritmo para obtener números pseudoaleatorios (ver imagen 4). La cantidad de números pseudoaleatorios que deben obtenerse debe ser igual al número de caracteres del mensaje a transmitir.

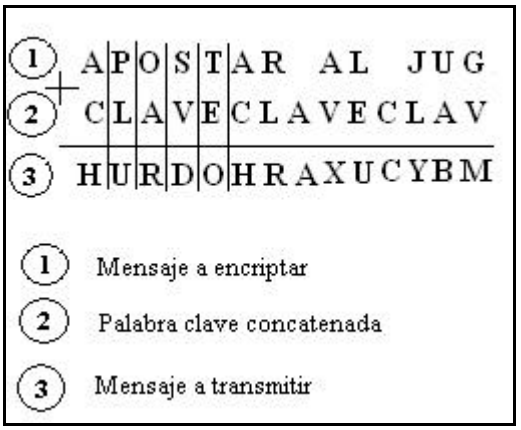


(Imagen 4: Generación de la semilla para el algoritmo de números pseudoaleatorios)

Una vez que se tienen los números pseudo-aleatorios se suma uno a uno el número del carácter a transmitir con los números obtenidos por el algoritmo. Luego de esto quedan nuevos valores, los que serán reemplazados en la tabla de valores por otros caracteres, los que finalmente se enviaran. En la otra punta se hará el proceso inverso. La palabra clave la deben tener los dos extremos para poder reproducir la cadena de números pseudo-aleatorios y poder obtener el mensaje original.
El defecto que tiene este tipo de encriptación es la generación de números pseudoaleatorios, porque se necesita que el algoritmo genere números con periodos largos, para que no se empiecen a repetir desde cierto momento.

Encriptación por palabra:

Este tipo de encriptación se basa en una palabra secreta, conocida únicamente por los extremos de la comunicación. Al igual que en el anterior método también cada carácter del alfabeto se hace corresponder con un número. Deben ser consecutivos también.
La palabra clave se concatena a si misma tantas veces como la longitud del mensaje a encriptar, luego se suma letra a letra con los valores de su correspondencia, obteniéndose nuevos caracteres que serán los que se transmitirán (ver imagen 5)



(Imagen 5: Encriptar por palabra clave)

Este tipo de encriptación hace imposible un ataque probabilística porque una letra puede ser representada de varias maneras, dependiendo con cual carácter de la palabra clave fue sumada. Para descifrar este tipo de encriptación se debe conocer la palabra clave y la correspondencia de caracteres – números.

Encriptación a mensaje ampliado:

En este método hace corresponder un carácter del texto plano a varios caracteres del texto a transmitir. El mensaje se amplia tantas veces como este programado el algoritmo, como por ejemplo, un carácter a cuatro. Como los anteriores usa también una correspondencia de carácter a números que debe ser secreta como las anteriores y consecutiva.
El algoritmo procede de la siguiente forma:

1- Se genera una correspondencia aleatoria idéntica a la de la figura 4.

2- Se generan los cuatro valores que corresponderán a caracteres de la siguiente forma:
   Num1 = Int(B + (64 - B) * Rnd)
   Num2 = Int(B * Rnd)
   r1 = Num1 + Num2
   Num3 = Int((r1 - B) * Rnd)
   r2 = Num1 + Num2 – Num3
   Num4 = r2 – B
   B = es el valor del carácter según la tabla secreta.
3- Hace corresponder los Num1, Num2, Num3 y Num4 con sus correspondientes caracteres de la tabla secreta.
4- Se sigue con el carácter siguiente.

De esta manera se crean cuatro números que se corresponden con caracteres de la tabla, de tal manera que:
car1 + car2 - car3 - car4 = CaracterOriginal
Cada carácter se puede formar de innumerables formas posibles por lo que complica mucho su descifrado.
El problema de este algoritmo es que agranda el mensaje, por lo que se necesita recursos para transmitir un mensaje pequeño.

Encriptación de archivos:

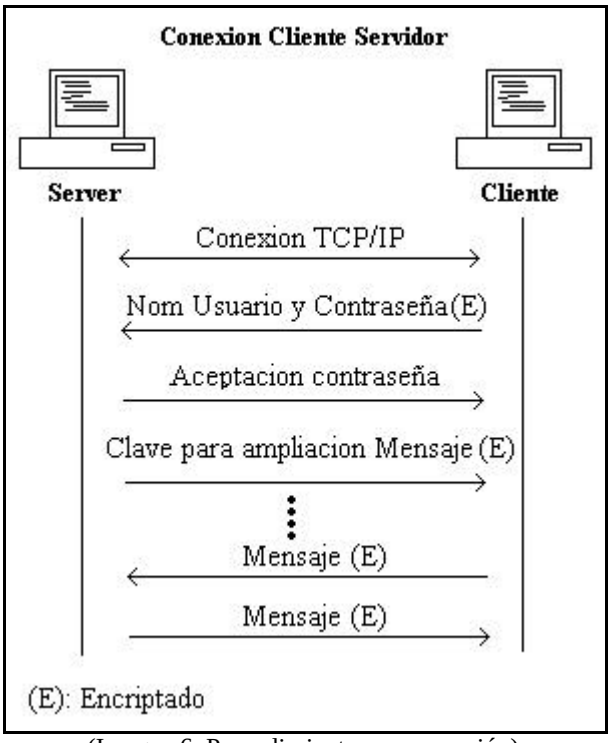
La encriptación de archivos no es muy diferente a la de los mensajes comunes. Se utiliza un alfabeto de 255 caracteres que representan a todos los bytes posibles. Se utiliza una clave aleatoria para hacer sustitución directa. También puede emplearse el método por palabra, o el de mensaje ampliado. Este ultimo es mas difícil y mas costoso aplicarlo, ya que para archivos grandes seria bastante complicado aumentarlos en 4 o mas veces, ya que el tiempo de transmisión aumenta radicalmente. El proceso de encriptación de archivos puede llevar a ser lento, dependiendo del tamaño del mismo, ya que el algoritmo toma cada byte y lo reemplaza con otro.

Aplicaciones en conexiones mediante Socket:

El trabajo presentado utiliza los métodos descriptos en una aplicación sencilla de cliente – servidor, con posibilidad de chat entre los clientes. Consta de una aplicación Server que controla las conexiones y maneja las claves de los usuarios y una aplicación Cliente.
El Server arranca y abre los puertos de escucha para aceptar peticiones de conexión. El cliente debe estar registrado en la base de datos del Server. Este pone su nombre de usuario y contraseña y presiona conectar.
Antes de mandar la contraseña se inicia la conexión TCP/IP exitosamente, luego el Server pide contraseña. En el lado del cliente se encripta la contraseña con el método de Número de Palabra, utilizando la propia contraseña como palabra clave y como mensaje a transmitir. Se mandan esos datos (el nombre de usuario no se manda encriptado), en el lado del Server se compara la contraseña encriptada con la de la base de datos, si resulta exitosa se logra la conexión.
En ese momento, en el lado del Server se ejecuta una algoritmo para generar una clave de correspondencia de letras-letras-números para enviar al cliente y que este utilice el método de mensaje ampliado cuando quiera comunicarse con el Server. La parte de letra-letra se necesitara para usar una sustitución directa previa, combinando dos métodos.
Una vez generada esa clave se la encripta mediante el método de encriptación por palabra, utilizando como clave la contraseña del cliente. De esta manera la clave viaja segura. Vea graficado el procedimiento en la imagen 6.
De ahora en mas cuando el cliente quiera mandar una mensaje al Server utilizara la clave que le ha sido asignada.
En el momento en que se quiera hacer uso del chat entre dos clientes, se procederá de la siguiente manera:

1- El cliente manda un mensaje con destino a otro cliente, este primero encripta el mensaje utilizando su clave y la manda al Server.

2- El Server recibe el mensaje y lo desencripta utilizando la clave del cliente de origen. Luego revisa el destino y encripta nuevamente el mensaje pero utilizando la clave del destino y manda el mensaje
3- El cliente destino recibe su mensaje y lo desencripta utilizando su propia clave.



(Imagen 6: Procedimiento para conexión)

De esta manera no hace falta una clave en común entre los clientes para que puedan dialogar, sino que se utiliza al Server como intermediario para realizar la conversación.
Todos los traspasos de mensajes entre los clientes y el servidor van encriptados de algunas de las maneras descriptas.
Supongamos que un intruso copia el primer mensaje en donde se pasa la contraseña encriptada y lo manda al Server como si fuera el propio usuario. Se podrá logear al Server, pero cuando el Server le mande la clave que vaya a usar, este intruso no lo podrá descifrar, ya que no conoce la contraseña, sino su representación encriptada y no podrá dialogar con el Server.

Resultados:

La utilización de estos métodos hace que una conexión sea segura en términos de privacidad de los datos y autenticidad de parte de los clientes.
De la manera descripta se utilizan todos los métodos en la comunicación para lograr que sea segura y a prueba de intrusos. Son métodos fáciles de implementar y que resultan eficientes para información no tan critica. Son un buen ejemplo para los que quieran diseñar aplicaciones seguras mediante conexiones en Socket. Incluso con un poco de imaginación, se pueden mezclar los métodos para que resulten incluso mas seguros y mas difíciles de quebrar.

Conclusión:

La criptografía se utiliza cada vez mas en nuestros tiempos, porque son cada vez mas frecuentes y mayores los peligros a los que se enfrenta nuestra información. Cada vez se utilizan métodos mas avanzados y se investiga nuevas formas de protegernos frente a los intrusos que intentan robarnos nuestros secretos. Estos métodos descriptos se pueden implementar en infinidad de aplicaciones, desde un simple chat, a programas avanzados que mandan un mayor caudal de información por la red, como aplicaciones de bases de datos distribuidas. Toda esa información puede asegurarse y podemos quedarnos tranquilos de que nadie excepto el destino leerá el mensaje. Mientras mas intrincado sean los métodos, o las combinaciones entre ellos mas se le dificultara la tarea a los que trataran de leerlo. Por eso se resumieron algunos métodos simples que pueden ser combinados de muchas formas para imposibilitar su lectura o su apertura si fuera un archivo.

Datos Personales:

Cargnelutti, Pablo Rubén.
Universidad Tecnológica Nacional
Facultad Regional Córdoba
Av. San Martín 970, Colonia Caroya, Córdoba (5223)
e-mail: pablocargnelutti84@hotmail.com
pabloruben1984@yahoo.com.ar