UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

“Criptoanálisis del algoritmo RC4”.

T E S I S

QUE PARA OBTENER EL TITULO DE

MATEMÁTICO

P R E S E N T A

Siddhartha Estrella Gutiérrez

DIRECTORA DE TESIS: M. en C. María de Lourdes Guerrero Zarco

MEXICO, D.F. Abril, 2006

Dedicatoria

Dedico este trabajo en primer lugar al Señor mi Dios por darme el discernimiento para acabar la carrera y salir adelante en la vida.

A mis padres que fueron y han sido un gran apoyo en todas las decisiones de mi vida.

A Tatiana mi Esposa por tener fe en mí sin condiciones así como ser mi musa inspiradora.

Agradecimientos:

A la Universidad Nacional Autónoma de México por brindarme la oportunidad de estudiar y abrir mi campo profesional de trabajo, tanto en la Facultad de Ciencias como en la Dirección General de Servicios de Computo Académico, especialmente al Departamento de Supercómputo. Así como a la Unidad Administrativa del Posgrado.

A mi padre por financiarme la carrera.

A mi madre por sus consejos desde que era niño

A mi directora de tesis M. en C. María de Lourdes Guerrero Zarco por todas sus atinadas propuestas y correcciones, sin ella esta tesis no hubiera sido lo que fue.

A mi Hermana Hypatia Deyanira por preocuparse de que hiciera mi tesis, gracias a tu libro pude darme idea de cómo armar todo.

A mi amigo Eduardo Murrieta Cabrera, con quien he podido realizar con su apoyo una buena programación y paralelización.

Al Doctor Octavio Páez Osuna quien me ha apoyado para seguir adelante en estas investigaciones.

Al Ing. Fernando Zaragoza Hernández por facilitarme algunos datos interesantes y muy importantes para la elaboración de esta tesis

Y a todas las personas que no puedo nombrar sin correr el riesgo de omitir alguna, por sus consejos, apoyos, y solidaridad conmigo para poder acabar mi carrera y posteriormente esta tesis.

Principio de Kerckhoffs

La seguridad de un criptosistema no debe depender en mantener secreto el algoritmo criptográfico. La seguridad depende solo de mantener secreta la llave

La cripotographie militaire (1883)
Alexandre Auguste Kerckhoffs Von Nieuwenhof

"No hay justicia ni injusticia, simplemente hay matemática. Y sobre todo, hay unas reglas. Los no iniciados no perciben las reglas, por que lo que perciben es desorden y caos. Mientras que el iniciado percibe la regla, la matemática divina, el orden, el arquitecto del universo, el geómetra..."

Las claves del Código Da Vinci, Lorenzo Fernández Bueno.

INDICE

RESUMEN

En este trabajo se realizó una investigación respecto a la seguridad de la información, se abordo el problema desde varias perspectivas; además se vio un algoritmo en concreto, el algoritmo RC4 que es utilizado ampliamente a nivel comercial.

Para este algoritmo existe software libre por lo que se desarrollaron programas en ANSI C que realizan ataques por diccionario sobre la contraseña a partir del texto cifrado partiendo que se conoce el texto en claro y se desea conocer la contraseña utilizada, siendo de interés el tiempo empleado para encontrarla.

De manera similar se ejecuta un ataque por fuerza bruta sobre la contraseña para conocer la contraseña empleada así como el tiempo que tardará en encontrar dicha llave.

Estos programas se ejecutaron en diferentes plataformas de cómputo con el fin de conocer que tan seguro es este algoritmo de cifrado respecto a la llave puesta; y bajo que plataforma es mas rápido obtener una llave (De ser posible esto).

Objetivo.

La importancia respecto a la seguridad de los algoritmos más utilizados que mantienen oculta la información de las personas, organizaciones así como instituciones es vital pues la información en este mundo globalizado y se convierte en un producto valioso e importante a nivel mundial que puede provocar grandes beneficios así como grandes pérdidas si esta información cae en las manos equivocadas [ACA05].

Cuando tratamos información confidencial debemos tener la certeza de que esta es realmente segura bajo los métodos estándar que se nos dan para mantener el secreto bajo una clave ya que el usuario común es lo que utilizará (el método estándar).

Existen una gran cantidad de algoritmos que nos permiten el cifrado de la información y RC4 es uno de los algoritmos más populares para hacer esto a nivel comercial.

El objetivo es analizar que tan seguro es el algoritmo estándar para mantener la confidencialidad de las aplicaciones sobre claves que utilizaría el usuario promedio en un cifrado de 40 bits en RC4 en base a la cantidad de tiempo empleado en romper la llave, además de saber si podemos confiar en una contraseña cualquiera o en una clave débil tratando de averiguarla por medio de un ataque por diccionario.

Hipótesis

Si tenemos un algoritmo fuerte; pero que al cifrar usa un número de bits bajo en la clave, este algoritmo se hace débil.

Si tenemos una contraseña débil aunque se tenga el más fuerte de los algoritmos, este algoritmo se hace débil.

INTRODUCCIÓN

Los sistemas de cifrado han sido uno de mis centros de atención desde que tenía 17 años, por lo que después de unos años me dedique a estudiar los métodos de cifrado, desde los algoritmos poli-alfabéticos hasta los que han surgido en nuestros días tales como los de clave simétrica y clave asimétrica o de llave pública y llave privada.

Estos temas han despertado en mí una inquietud para saber qué tan segura viaja la información. Los chinos de la antigüedad así como los egipcios tenían un buen método para mantener secreta su información, pues muy poca gente podía leer lo que estaba escrito. En nuestros días la seguridad por oscuridad, es decir la seguridad por no saber lo que algo dice o hace ya no es funcional; dada la cantidad de información que se tiene disponible tanto en libros de bibliotecas públicas como en la Internet.

Con ello surge la pregunta: ¿Qué tan seguros son los métodos utilizados actualmente para poder cifrar la información?

Existen una gran variedad de métodos así como una gran cantidad de aplicaciones y algoritmos para cifrar la información, entre ellos DSA, triple DES, Diffie-Hellman, etc. Pero el algoritmo que se analizará en el presente trabajo es el algoritmo RC4, que cuenta con una gran cantidad de aplicaciones inclusive para comunicaciones de redes inalámbricas, además de ser utilizado para el cifrado de documentos de Microsoft Word, implementado en openssl que es un software libre que sirve de plataforma para muchos otros programas libres para linux tales como apache, postfix, servidores pki y muchas otras aplicaciones más.

Además RC4 tiene un cierto toque de misticismo puesto que inicialmente fue un software propietario que finalmente fue dado a conocer al público en una lista de manera anónima.

Comenzaremos entonces con la investigación, la intención es que en base al cálculo de la complejidad del algoritmo que ya se tiene calculada y no es la intención de este trabajo realizar, hacer la propuesta de un criptoanálisis del algoritmo mediante un ataque por fuerza bruta además de un ataque por diccionario, estos dos ataques se harán conociendo de antemano el texto en claro o una parte del texto en claro, esta implementación de la herramienta utilizará openssl, ejecutándose bajo un sistema linux esto es que se ejecutará la herramienta en software libre en la manera de lo posible. Además

se realizará un ataque por fuerza bruta sobre el algoritmo, en el transcurso del trabajo se explicará detalladamente en que consisten estos dos tipos de ataques así como cual es la diferencia entre ellos.

Mucho se habla de la debilidad de las contraseñas por parte de los usuarios, en este trabajo se harán pruebas de contraseñas por medio de un ataque por diccionarios así como ataques por fuerza bruta sobre el algoritmo RC4 a 40 bits para probar un algoritmo seguro con contraseñas inseguras, se asegura que RC4 a 40 bits también es vulnerable y se verá cual es la razón de esta afirmación, aunque cabe señalar que RC4 a 1024 bits es una opción bastante confiable para mandar información cifrada a través de cualquier medio inseguro.

Capítulo 1

1.1 Definición de criptografía

La Real Academia Española define criptografía (oculto + escritura) como:
"El arte de escribir mensajes con una clave secreta o de modo enigmático".

Aunque esta definición es interesante [RAM04], resulta ser muy poco apegada a los tiempos modernos.

La criptografía es ambas cosas: el candado y la combinación (Llave), que protege un mensaje secreto de cualquiera que no conoce la llave.

¿La criptografía es un arte? La criptología ha dejado de ser un arte para convertirse en una ciencia. Además no sólo se escriben mensajes; se envían o se guardan en una computadora diversos tipos de documentos con distintos formatos (txt, doc, exe, gif, jpg, etc) y los sistemas actuales usan una o dos claves.

En la antigüedad se usaba una clave secreta, pero actualmente existen sistemas de clave secreta que usan una sola llave y sistemas de llave pública (muy importantes) que usan dos: una llave privada (secreta) y la otra pública.

En cuanto a la representación enigmática; la representación binaria de la información podría ser enigmática para nosotros los humanos pero no para los computadores pues es su lenguaje natural, pues las computadoras solo entienden ceros y unos, ellas no entienden palabras tal y como nosotros las entendemos sino que es un lenguaje diferente que se compone de ceros y unos. Lo que para ellas es natural, nosotros no lo podríamos entender, en cambio aunque es un lenguaje que esta compuesto de ceros y unos, una computadora con las mismas características podría entender el mismo lenguaje sin necesidad de un software o hardware especial por lo que esta información no está segura, aunque si es un enigma para un humano pues no sabría interpretarla.

Lo que realmente es la criptografía moderna:

Rama inicial de las Matemáticas y en la actualidad de la Informática , que hace uso de métodos y técnicas con el objeto principal de cifrar y/o proteger un mensaje o archivo por medio de un algoritmo, usando una o más llaves [RAM04]. Esto da lugar a diferentes tipos de sistemas de cifrado que permiten asegurar cuatro aspectos de la seguridad informática: la confidencialidad, la integridad, la disponibilidad y el no repudio de emisor y receptor.

El criptoanálisis es construir un análisis lingüístico y matemático que remueve el enmascaramiento creado por criptógrafos.

Se tiene el modelo teórico de lo que es un criptosistema con todas las condiciones que debe tener como se muestra en la figura 1.1

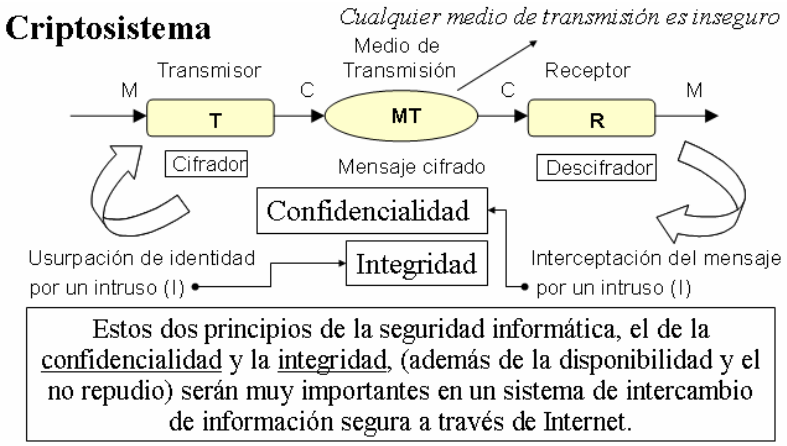


Fig. 1.1 Criptosistema

Criptografía es construir una cripta, en donde la gente tradicionalmente ha escondido cosas de valor para mantenerlas a salvo en el camino al mas allá, esta palabra viene del griego Kriptos que significa oculto ó cubierto, del Nórdico antiguo hreysar, significa montón de piedras, y el lituano krauti, significa apilar. Es la forma de ocultar la escritura (“Grafía”) pero mantener la manera de encontrarla de nuevo, como apilar piedras en una lápida sepulcral a la cual se quiere regresar. [BAK00]

La criptografía es la ciencia de escribir en secreto [SAN03], que ayuda a comunicarse sin revelar la información a los adversarios, además de proteger nuestras identidades. Por medio de esta ciencia se pude proteger cualquier tipo de información, incluso datos muy importantes, tal como información generada en comercio electrónico y transacciones bancarias. Pero aunque es una herramienta muy poderosa se debe ser cuidadoso ya que si no se maneja de manera responsable y bien aplicada, se puede tener un falso sentimiento de seguridad. La criptografía debe ser parte de una larga y profunda estrategia de seguridad, y esta debe proveer solo una de las capas de la seguridad.

1.2 Definición de cifrado

La palabra cifrar se obtiene de la raíz árabe "sifr" y cero, después del siglo XIII de nuestra era, los Europeos comenzaron a utilizar los numerales árabes preferentemente a los romanos, porque en el medioevo los decimales y los ceros facilitaron las cosas a los matemáticos. Pero el concepto del cero confundía a la gente común de la edad media, así que cuando ellos se referían a algo que no era bastante claro, lo comparaban con algo que era considerado un misterio: "el cifrado". Con el tiempo la palabra cifrar sirvió para describir el ocultamiento de algo en claro. [BAK00]

El uso del cifrado en el mundo se remonta a los egipcios hace 4000 años atrás, además en el siglo XX juega un papel crucial en ambas guerras mundiales. [MEN96] Los practicantes de este arte fueron principalmente las personas relacionadas con la milicia, la diplomacia y el gobierno en general. La criptografía fue usada como una herramienta para proteger secretos nacionales y estrategias.

La masificación en el uso de computadoras y sistemas de comunicación en los años 60 se extendió hasta el sector privado y se tuvo la necesidad de mantener segura la información que estaba en forma digital y proveer estándares de seguridad. La respuesta la dio Feistel de IBM a principios de los años 70 y culmino con la adopción por parte de los Estados Unidos del U.S. Federal Information Processing estándar para el cifrado de información no clasificada. El DES (Data Encription Standard) es el mecanismo de cifrado más conocido en la historia. Fue el estándar para mantener la seguridad del comercio electrónico de muchas instituciones financieras alrededor del mundo.

Feistel quería llamar a su sistema Dátasela, pero IBM acortó el termino demostración Chiper por Demon. Posteriormente Demon cambio a Lucifer, el cual fonéticamente contenía la palabra cifrado (cipher). Al final el nombre degeneró en DES.

RSA Data Security, llamada así después de la invención del algoritmo RSA de cifrado de llave pública, patrocinó el primer Reto DES en enero de 1997. El premio al primer lugar fue para Rocke Verser quien rompió el DES recobrando la llave secreta en 96 días. Menos de un año después, en febrero de 1998 un grupo de Distribuited.net rompió el DES en 41 días.

Cuatro meses después, en julio de 1998 un grupo de la Electronic Frontier Foundation (EFF) y de la Distribuited.net utilizaron una máquina valuada en menos de

$250,000 USD, y rompieron DES en 56 horas. Medio año después en enero de 1999, el mismo equipo completo la misma tarea en menos de 24 horas, DES ya no era un algoritmo seguro.

A través de los años se han elaborado protocolos y mecanismos que han sido creados específicamente para mantener segura la información en documentos físicos. A menudo la seguridad de la información no solo se debe dejar a los algoritmos matemáticos sino también se debe tener una legislación al respecto para tener el resultado deseado es decir, se deben tener leyes que regulen y legislen el uso de algoritmos y se castigue el intento no autorizado de poner en claro la información cifrada.

En la figura 1.2 se muestra el proceso de cifrado y descifrado, primero se tiene el texto base a cifrar que puede ser un mensaje, un número de cuenta bancaria o información confidencial, este texto es cifrado mediante algún algoritmo para su posterior transmisión por medio de un canal inseguro, este mensaje viaja cifrado y si es interceptado; dicho mensaje no es entendible por el intruso, el mensaje llega a su destinatario quien por medio del algoritmo de descifrado obtiene el texto base. Se tiene el espacio de mensajes que se pueden enviar que, al ser cifrados este es parte del espacio de textos cifrados, para cifrarlo se toma una clave que pertenece al espacio de claves que realiza al enviar y recibir las transformaciones de cifrado y descifrado.

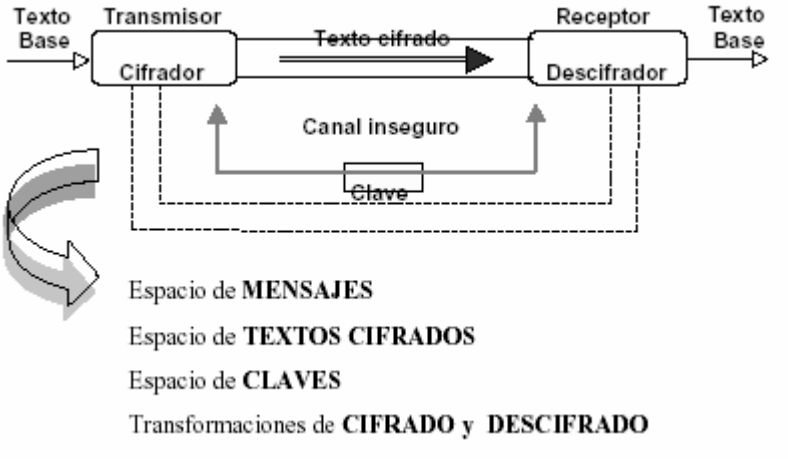


Figura 1.2

1.3 Por qué utilizar cifrado

El cifrado es de vital importancia para asegurar información. Uno de los objetivos principales es ayudar a defendernos de las personas que escuchan las comunicaciones ajenas sin permiso. La idea es que la comunicación sobre cualquier tipo de medio tiene un riesgo inherente de que un tercero pueda estar escuchando la comunicación de manera no autorizada y se desea reducir el riesgo de intercepción de la información y su difusión a personas no autorizadas. De esta manera el cifrado hace que el texto interceptado sea inentendible para el interceptor.

Comúnmente un algoritmo de cifrado realiza dos tareas distintas: el cifrado y el descifrado. Cifrar es la práctica de disfrazar el mensaje de tal manera que es imposible de entender. El mensaje transformado depende matemáticamente de una fórmula llamada algoritmo de cifrado. Una vez que el mensaje ha sido transformado con un cifrador, el mensaje que resulta es llamado texto cifrado el cual es ilegible. Para que el mensaje pueda ser leído en su forma original este deber ser descifrado. El descifrado es el proceso de transformar un mensaje cifrado de vuelta a forma original de texto en claro o texto ordinario.

Pero ¿Quiénes crean los algoritmos?

Los algoritmos son creados por científicos llamados criptógrafos, que dominan varios campos de las matemáticas y comúnmente trabajan en grupos, toma muchos años inventar y refinar algoritmos de cifrado. Pero en gran dependencia de la criptografía se encuentra el criptoanálisis, los criptoanalistas dedican sus vidas a romper códigos cifrados. Algunos criptoanalistas trabajan para la milicia y el gobierno, otros solo se interesan en el estudio de las debilidades de los algoritmos para asegurarse que no puedan ser rotos por otros. El término genérico para el estudio de ambos tanto criptografía como criptoanálisis se llama criptología. [SAN03]

1.4 Cifrado y comercio electrónico

En el comercio electrónico, comunicaciones militares, del gobierno, y en comunicaciones diplomáticas se exigen comunicaciones seguras. El cifrado es una de las tecnologías esenciales para el comercio [SAN03]. En particular el cifrado ayuda a asegurarse que:

- Las comunicaciones con el servidor correcto no sean atacadas mediante "*ip spoofing*" (Clonar la dirección ip) por un impostor.
- Los mensajes no puedan ser alterados sin previo conocimiento del que lo envía
- Los mensajes que se envían realmente son entregados
- Se pueda probar que alguien mas (Un tercero que es intruso) no envió mensajes que los participes de la comunicación enviaron.
- Solo los interesados en la comunicación puedan leer el mensaje.

De manera similar el cifrado ayuda a los vendedores de comercio electrónico que:

- Se están comunicando con el cliente correcto y no con un impostor.
- Los contenidos del mensaje recibido son correctos e inalterados.
- No existe duda sobre la identidad de la persona que envía el mensaje.
- Se puede asegurar que el creador del mensaje sea realmente quien lo ha enviado por medio de niveles de confianza mediante un protocolo de cifrado, es decir; mediante algún algoritmo de autenticación.

1.5 Requisitos de un criptosistema

Los requisitos para un criptosistema son: [RAM04]

- Algoritmo de cifrado/descifrado rápido y fiable, en donde rápido se entiende como el poder cifrar y descifrar en un tiempo practico razonablemente corto y fiable se entiende como que realiza un cifrado libre de errores que puedan, a la hora de hacer el cifrado o descifrado acarrear una mala interpretación del mensaje ya sea cifrado o como texto base.
- Posibilidad de enviar archivos por una línea de transmisión, almacenarlos o transferirlos de un lugar a otro.
- No debe existir retardo debido al cifrado o descifrado.
- La seguridad del sistema deberá residir solamente en el secreto de una clave y no de las funciones de cifrado tomando en cuenta el protocolo de cifrado es decir; dependiendo del algoritmo de cifrado es necesario seguir ciertas reglas que me permitan no detectar de manera sencilla las reglas del algoritmo ó detectar posibles colisiones así como el tipo de algoritmo utilizado.

- La fortaleza del sistema se entenderá como la imposibilidad computacional de romper el cifrado o encontrar la clave secreta.

1.6 Objetivos de la criptografía:

Se pueden sintetizar los objetivos de la criptografía de la siguiente manera [MEN96]:

1. Confidencialidad. Es el servicio que mantiene el contenido de la información de manera no legible por medio de algún algoritmo matemático.
2. Integridad de los datos. Es el servicio que delata la alteración no autorizada de información. Asegura la integridad de los datos una vez que se tiene la capacidad de detectar los cambios de la información por personal no autorizado o por errores de transmisión.
3. Disponibilidad permite que los datos se utilicen cuando sea necesario. Que esten al alcance de sus usuarios y destinatarios y que se pueda acceder a ellos en el momento en que se necesitan utilizar.
4. Autenticación. Servicio en el que se proporcionan los datos relacionados con la identificación tanto del emisor como del receptor.
5. No rechazo (No repudio). Es el servicio que previene a una entidad el negar previamente entregas o acciones.

Además definiciones equivalentes son las siguientes: [BOR01]

La privacidad o confidencialidad es la necesidad de que dicha información solo sea conocida por personas autorizadas. Ya que en caso contrario la información podría provocar severos daños a su propietario (Por ejemplo la caída inminente de la bolsa de valores el Miércoles de la próxima semana), o volverse obsoleta (Por ejemplo la próxima generación de televisores de conocida compañía en manos de sus competidores)

La integridad de la información es la característica que hace que su contenido permanezca inalterado a menos que sea modificado por personal autorizado, y esta modificación sea registrada para posteriores controles o auditorias. Una falla de integridad puede estar dada por anomalías en el hardware, software, virus informáticos y/o modificación de personas que se infiltran en el sistema.

La Autenticidad permite definir que la información requerida es válida y utilizable en tiempo, forma y distribución, validando al emisor de la misma para evitar suplantación de identidad.

El no repudio con el que se evita que cualquier entidad que envió o recibió información arguya ante terceros que no la envió o la recibió.

Agregando además los siguientes aspectos [BOR01]:

- Protección a la réplica. En esta se asegura que una transición puede realizarse solo una vez, a menos que se especifique lo contrario. No debe poderse grabar una transacción para luego reproducirla, con el propósito de copiar la transacción para que simule que se recibieron múltiples peticiones del remitente original.
- Control sobre la información permite asegurar que solo tendrán acceso a la información personas autorizadas, además de establecer una política para permitir el acceso autorizado.
- Consistencia: Se debe poder asegurar que el sistema se comporte como se supone debe hacerlo, ante los usuarios que corresponda.
- Auditoria: Es la capacidad de determinar que acciones o procesos se están llevando a cabo en el sistema, así como quien y cuando los realiza.

1.6.1 Amenazas.

Una amenaza se define como cualquier elemento que comprometa el sistema [BOR01], en la Figura 1.3 se muestran los principales tipos de amenazas.

Figura 1.3

Las amenazas pueden ser catalogadas y analizadas en tres momentos: antes, durante y después del ataque. Estas 3 modalidades tienen diferentes tipos de políticas que deben garantizar la seguridad de un sistema informático [BOR01].

- Prevención: Mecanismos que aumentan la seguridad de un sistema durante su funcionamiento normal. Por ejemplo el cifrado de la información para su transmisión.
- Detección: Mecanismos orientados a revelar violaciones a la seguridad. Tales como programas de detección de intrusos o programas de auditoria.
- Recuperación: Mecanismos que se aplican, cuando la violación del sistema ya se ha detectado, para retomar éste funcionamiento normal. Por ejemplo la recuperación de una base de datos a partir de una copia de seguridad.

La manera ideal para enfrentar las amenazas que se enfrentan al momento de asegurar la información son los mecanismos de prevención, por lo que debe hacerse un análisis de lo que se quiere proteger, con que medios cuento para protegerlo y por cuanto tiempo debe protegerse [ACA05].

En la figura 1.4 se muestra la estructura de un criptosistema, en él se tiene un mensaje, un medio transmisor y un receptor, esta trayectoria del mensaje es susceptible a varios ataques por ejemplo alguien ajeno puede interceptar el mensaje por lo que la confidencialidad debe estar presente y así el intruso no podrá leer el mensaje, además el intruso puede modificar el mensaje independientemente de si lo pudo leer o no, por lo que es importante cerciorarse que los datos están íntegros, es decir que nadie ajeno al creador de la información ó a las personas autorizadas ha modificado los datos, se debe saber tanto quien envió el mensaje como de quien lo recibió, y por último se debe tener la seguridad de que el mensaje que se envía llegará sin ningún problema a su destinatario, es decir que no será filtrado de alguna manera o eliminado antes de que llegue a su destino.

En la figura 1.4 se muestra la inseguridad del medio de transmisión, en el que se puede infiltrar un tercero y escuchar la comunicación modificándola de alguna manera entre las flechas que indican “C” que es la comunicación entre “T” y “R”

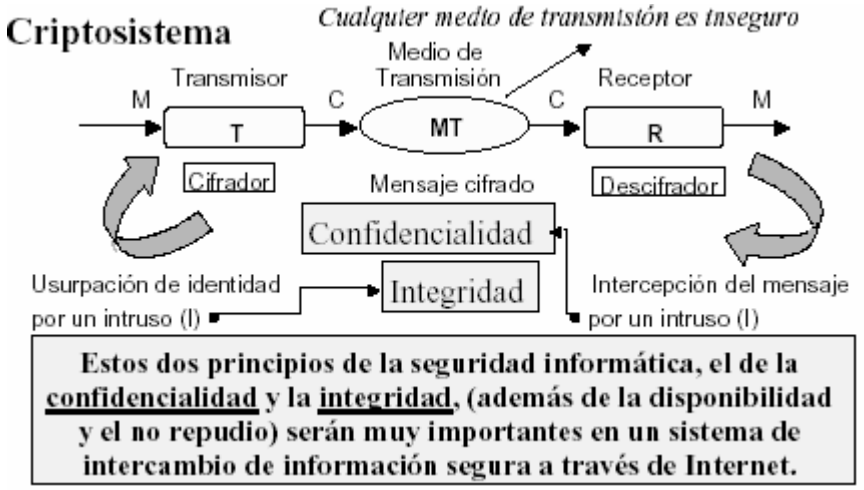


Figura 1.4

Un objetivo fundamental es direccionar adecuadamente estas 4 metas tanto en la teoría como en la práctica (Integridad, Confidencialidad, Autenticación, No repudio). La criptografía se involucra en la prevención de engaños así como de actividades maliciosas.

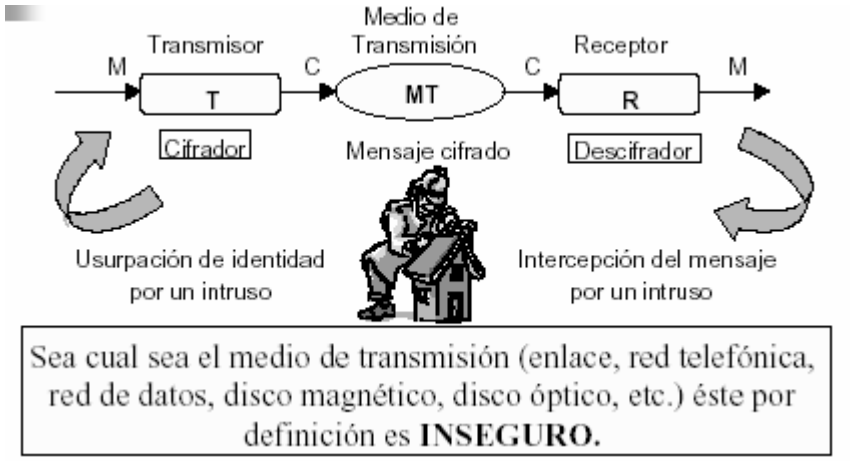
1.7 Criterios de evaluación de un algoritmo de cifrado

Un algoritmo de cifrado debe ser evaluado con respecto a varios criterios tales como [MEN96]:

1. Nivel de seguridad. A menudo es dado por el número de operaciones requeridas (Utilizando los métodos actualmente conocidos) para vencer el objetivo propuesto.
2. Funcionalidad. Los algoritmos necesitarán ser combinados para conjuntar los objetivos de la seguridad.
3. Métodos de operación Los algoritmos, cuando se aplican de varias maneras y con varias entradas, típicamente exhibirán varias características diferentes, así un algoritmo puede proveer varias características dependiendo de su modo de operación.
4. Desempeño Esto se refiere a la eficiencia del algoritmo en un modo particular de operación (Por ejemplo un determinado bloque de bits por segundo que el algoritmo puede cifrar)
5. Fácil implementación. Esto se refiere a la dificultad de realizar el algoritmo en la implementación práctica. Esto pudiera incluir la complejidad de la implementación del algoritmo en un ambiente de software y hardware determinado.

La importancia de los criterios está en relación con la aplicación de algoritmos y los recursos disponibles. La criptografía en los últimos 30 años ha tenido un período de transición como disciplina, transformándose de arte a ciencia.

En la Figura 1.5 se destaca la importaría del uso de un algoritmo de cifrado pues cuando el mensaje viaja por el medio de transmisión este es por naturaleza, inseguro ya que este medio comúnmente puede ser monitoreado por un sniffer como en el caso de la Internet aunque estén diversas opciones para evitar el paso de los datos en claro como en el caso de IPsec que es la base para las redes virtuales (VPN), pero en algunos correos electronicos gratuitos estas opciones no son activadas o por ejemplo en los chats como messenger ó icq y aplicaciones por el estilo.



C = Espacio de textos cifrados

M = Espacio de mensajes

Figura 1.5 Ataque del Hombre en Medio

1.8 Definición de un cifrador por flujo.

Existen dos formas de manipular información mientras se cifra y descifra; se puede romper la información en bloques o se puede cifrar el flujo bit por bit. En este caso los esquemas de cifrado son clasificados en cifradores por flujo o cifradores por bloque, dependiendo de cuanta información se genere a la vez y de cómo se genera la llave [RIV04].

Cuando el algoritmo no tiene que dividir el mensaje sino que lo cifra todo completo (todo el bloque de información) se dice que cifra un flujo.

Usa el concepto de cifra propuesto por Vernam, que cumple con las ideas de Shannon sobre sistemas de cifrado secreto perfecto, esto es [RAM04]:

a) El espacio de las claves es igual o mayor que el espacio de los mensajes.

b) Las claves deben ser equiprobables, es decir cada posible clave debe tener la misma probabilidad.

c) La secuencia de llave se usa una sola vez y luego se destruye (sistema one-time pad).

Además:

- El mensaje en claro se leerá bit a bit.
- Se realizará una operación de cifrado, normalmente con la función XOR, con una secuencia cifrante de bits

También debe cumplir ciertas condiciones:

- Un período muy alto. (Es decir una secuencia larga de cifrado cuando es por bloque para que no se pueda reconocer un patrón o secuencia relativamente fácil de seguir en un mensaje cifrado)
- Aleatoriedad en sus propiedades.

En la Figura 1.6 tenemos un ejemplo del uso de un algoritmo al ser cifrado, el algoritmo en cuestión es de las más sencillos que existen para la práctica.

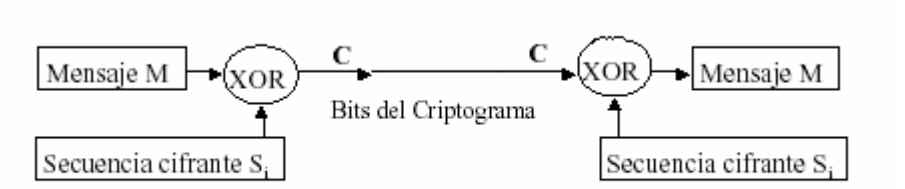


Figura 1.6

Además se da la siguiente definición de cifrador por flujo [SAN03]:

El cifrador por flujo opera en un bit simple, en un byte o en una palabra computacional a la vez y es implementado de alguna forma por un mecanismo alimentador (feedback) de ahí que la llave este cambiando constantemente. De manera ideal la llave en un cifrador de flujo es al menos tal grande como lo es el texto cifrado.

Un flujo de la llave (Keystream) es primero generado para enviar y recibir términos. Ambos términos deben mantenerse sincronizados uno con el otro y producir flujos de la llave idénticos (identical keystreams). Los flujos de la llave deben ser impredecibles por un observador externo, por lo que los flujos deben utilizar llaves. Al enviar términos, el flujo de la llave y el texto base son transformados (por ejemplo utilizando XOR) para producir un flujo del texto cifrado que es transmitido. Al recibir términos, el flujo de la llave idéntico (keystream) es extraído del flujo del texto cifrado, generando nuevamente el flujo del texto original. Los cifradores de flujo son muy dependientes de la aleatoriedad del flujo de la llave y son vulnerables al ruido durante la transmisión.

Mientras existe una variedad de cifradores de flujo, se mencionan dos genéricos. Un cifrador de llave automática o auto sincronizado que calcula cada bit en el flujo de la llave como una fusión de los N bits previos del flujo de la llave. Tiene la capacidad de mantener el proceso del descifrado sincronizado con el proceso de cifrado solamente por saber que tantos N bits de la llave de flujo lleva recorrida. Aunque un problema es el error de propagación; un error en un bit de transmisión se transforma en N bits basura al usuario final.

Están además los cifradores de flujo síncronos que generan un flujo de llave de manera independiente al flujo del mensaje utilizando la misma función que usa el flujo de la llave en ambos términos. Es importante que la función generadora de la llave sea impredecible a los ojos de un intruso.

1.9 Definición de un cifrador por bloque

La mayoría de los esquemas de cifrado utilizan el cifrador por bloque, lo que significa que se cifra un pedazo uniforme de información a la vez. Los cifradores por bloque pueden operar en uno o varios módulos [SAN03]. El modo en que se selecciona el bloque a cifrar afecta directamente la fortaleza y desempeño del criptosistema. Los siguientes cuatro modos son los más importantes:

- El modo de código de libro electrónico (EBC) es el más simple, la aplicación mas obvia; la llave es utilizada para cifrar el texto plano del bloque para formar un bloque de texto cifrado. Dos bloques idénticos de texto plano siempre generarán el

mismo bloque cifrado. Además este es el modo del cifrador por bloques más común y es susceptible a varios ataques por fuerza bruta.

- El cifrador cambiador de bloques (CBC) agrega un mecanismo de alimentación al esquema de cifrado. En CBC, el texto plano es tratado con XOR con el bloque del texto previo que se cifró. De este modo dos bloques idénticos de texto plano nunca se cifraran de la misma manera.
- El cifrador alimentador (CFB) es un cifrador por bloques que en sí mismo es un cifrador por flujo. El modo CFB permite que la información sea cifrada en unidades pequeñas del tamaño de un bloque, el cual es utilizado en algunas aplicaciones, tales como el cifrado interactivo de una Terminal de entrada. El modo a un byte por ejemplo hará el cifrado caracter por caracter en un registro tomándolo al tamaño de el bloque, cifrando el carácter y enviándolo al bloque. El usuario final recibe el bloque que es descifrado y los bits sobrantes que rellenaban el bloque son descartados.
- El modo de alimentación de Salida (OFB) es una implementación conceptualmente similar al cifrador de flujo síncrono. El OFB previene el mismo texto plano por bloque que genera el mismo texto cifrado por bloque pero utilizando un mecanismo interno de alimentación que es independiente de ambos, el texto plano y texto cifrado en flujo de bits.

Los cifradores por bloque pueden ser implementados como cifradores de flujo y viceversa; la diferencia esta en como se aplica el cifrado. Si se tiene un dispositivo de hardware, tales como una red virtual privada (VPN), los cifradores de flujo son fáciles de implementar vía hardware y puede ser la manera ideal de hacerlo, especialmente para flujos continuos que no terminan, tales como las ligas de comunicación. Si el cifrado se complementa con software, tal como el cifrar un archivo, el cifrador por bloque será mucho mas eficiente. Para implementar un software de cifrado por flujo se requiere de una gran cantidad de bits para la máscara, que puede resultar en errores de programación ó un bajo desempeño del algoritmo.

Con un cifrador por bloque, el texto plano es roto en bloques de longitud fija (comúnmente a 64 bits) y se procesa un bloque a la vez. Cuando es necesario, un último bloque puede ser agregado. Una transformación fija (el mismo algoritmo y la misma llave) son aplicados a cada bloque. Comúnmente un gran número de operaciones son ejecutadas

por cada bloque. Para la mayoría de los algoritmos la misma llave es utilizada para cifrar o descifrar cada uno de los bloques.

1.10 Ventajas y desventajas de un cifrador por bloque y un cifrador por flujo

Haciendo un estudio y comparación de ventajas y desventajas entre los cifradores por bloque y los cifradores por flujo se tiene lo siguiente:

1.10.1 Cifrado en bloque

Ventajas:

- Alta difusión de los elementos en el criptograma.
- Imposible introducir bloques extraños sin detectarlo.

Desventajas:

- Baja velocidad de cifrado al tener que leer el bloque.
- Propenso a errores de cifrado. Un error se propagará a todo el bloque.

1.10.2 Cifrador en flujo

Ventajas:

- Alta velocidad de cifrado al no tener en cuenta otros elementos.
- Resistente a errores. Cifra independiente cada elemento.

Desventajas:

- Baja difusión de elementos en el criptograma.
- Vulnerable. Pueden alterarse los elementos por separado.

1.11 El sistema ideal para alcanzar los cuatro puntos principales de seguridad en el cifrado.

Existe un modelo para alcanzar el nivel ideal de los cuatro puntos principales del cifrado, estos modelos son utilizados para satisfacer políticas en mensajes enviados y recibidos para cuestiones bancarias y de negocios, aplicaciones militares y diplomáticas como las que maneja la ONU, la OEA, y los ejércitos de todo el mundo, incluyendo el Ejercito Mexicano, mantener una comunicación segura , sin repudio, manteniendo la integridad del mensaje además de su confidencialidad , conociendo exactamente quien manda la comunicación (autenticación); requiere no solo de un algoritmo de cifrado, sino también de otros algoritmos también criptográficos que realizan esas tareas.

La imagen que se muestra a continuación permite observar la capacidad completa para cumplir los diferentes puntos que requiere un mensaje cifrado seguro, así como las herramientas necesarias para lograrlo, entre ellas la firma digital que es información añadida a través de una transformación cifrada de los datos que permite al receptor de los mismos comprobar su fuente e integridad y protegerse así de la suplantación o falsificación. Esta consiste en una transformación de un mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posea el mensaje inicial y la clave pública del firmante, pueda determinar de forma fiable si dicha transformación se hizo utilizando la clave privada correspondiente a la clave pública del firmante, y si el mensaje ha sido alterado desde el momento en que se hizo la transformación. Es un sello integrado en datos digitales, creado con una clave privada, que permite identificar al propietario de la firma y comprobar que los datos no han sido falsificados. Aunque estas herramientas pueden variar de una organización a otra, ya que existen diferentes aplicaciones en el mercado que hacen la misma tarea dependiendo de las necesidades de la información a tratar, existen herramientas libres y herramientas comerciales para hacer las tareas específicas, con openssl [COX05] que es una herramienta libre en la que se pueden realizar cada una de las tareas que se proponen en la imagen 1.7.

Es importante mencionar que las máquinas involucradas en la emisión y transmisión deben estar protegidas para mantenerse seguras, ya que si alguna de estas máquinas es comprometida ya sea de manera física o remota, entonces no se puede garantizar la confidencialidad del mensaje ó alguno de los 4 puntos principales pues no hay garantía de

que la maquina comprometida este en condiciones de cumplir los requisitos pues por algún programa instalado que sea malicioso puede dejar de cumplir las condiciones deseadas.

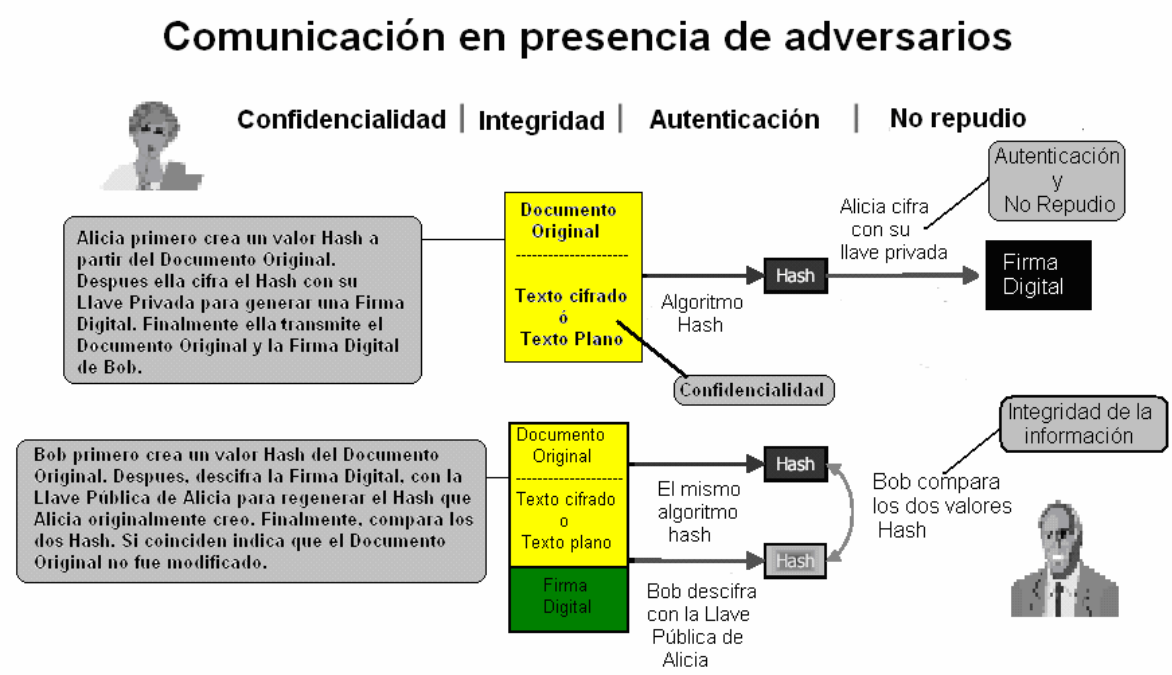


Figura 1.7

En la figura 1.7 se muestra como se deben cumplir la confidencialidad, la integridad, la autenticación, y el no repudio, de manera práctica a través de varias herramientas de cifrado.

Los ataques son algo común a los sistemas de información y no se debe confiar solamente en un algoritmo de cifrado por lo que se tienen que cumplir una serie de condiciones tanto en la forma en que se cifran los archivos, como en la manera que se guardan , pero es importante también , mantener las máquinas en donde se guarda la información de manera segura, por lo que en el capitulo 2 se dan una serie de sugerencias de medidas a tomar al respecto, no solo para el algoritmo RC4 sino también para cualquier algoritmo de cifrado. Debe recordarse y mantenerse siempre presente que la seguridad es un proceso de varias capas y el cifrado es solo una de ellas.

Capítulo 2

2.1 Controles de acceso a la información

Los controles de acceso a la información pueden ser implementados de varias maneras una de ellas es a través del sistema operativo otra sería por medio de una aplicación que cifre el archivo como el caso de pgp (Pretty Good Privacy) que es un algoritmo de cifrado, o por medio de alguna aplicación del mismo estilo pero que cifre el archivo con un algoritmo a nuestro gusto respecto al método de cifrado que queramos elegir y otra opción sería por medio de la misma aplicación que estamos utilizando para crear y editar un archivo por ejemplo Microsoft Word.

Todas ellas ayudan a proteger la privacidad de los accesos al sistema operativo, a las aplicaciones o a la información de accesos no autorizados o de modificaciones maliciosas.

Para hacer todo esto posible en tiempo y forma necesitamos un estándar que cubra las necesidades y las formas en que se deben autentificar los usuarios para tener acceso al sistema operativo, a las aplicaciones o a la información, por lo que el Nacional Institute for Standard and Technology (NIST por sus siglas en ingles) [NIS04] ha resumido los estándares de seguridad que se refieren a los requisitos mínimos de seguridad en cualquier sistema los cuales se explican en la siguiente sección.

2.1.1 Control de acceso a usuarios

Es la primera medida de seguridad para los sistemas de cómputo que previene el ingreso al sistema de personas no autorizadas además de ser la base que permite el uso de bitácoras para detectar las actividades y frecuencias de acceso y uso de los usuarios.

El usuario se identifica por medio del nombre de su cuenta, de la que el sistema solicitara una contraseña para verificar que el usuario es quien dice ser.

Existen varias técnicas que permiten verificar la autenticidad del usuario entre ellas se destacan 4 principales las cuales pueden ser independientes o combinadas entre si [NIS04]:

1. Algo que sólo conoce el individuo o ente a autenticar, por ejemplo una contraseña, una llave criptográfica, o un número personal.
2. Algo que la persona posee, como una tarjeta inteligente, un llavero con chip ó una tarjeta magnética.
3. Algo que el individuo es y que lo identifica de manera única con respecto a los demás; como las huellas digitales, la retina de algún ojo ó la voz.
4. Algo que el individuo es capaz de hacer, como los patrones de escritura.

Cada una de estas técnicas de autenticación tiene sus ventajas y sus desventajas, por ejemplo a veces las contraseñas se olvidan o las huellas digitales se dañan por algún accidente, las tarjetas se pueden dañar al caerse o el patrón de escritura puede cambiar con el paso de los años, pero también todas ellas tienen la ventaja de que no se va a poder entrar a un sistema si no se posee el medio de autenticación para acceder al sistema.

Además la seguridad informática se basa en la administración efectiva de los permisos de acceso de los recursos informáticos basados en la identificación, autenticación y autorización de accesos. La administración de estos permisos abarca [BOR01]:

1. El proceso de solicitud, establecimiento, manejo, seguimiento y cierre de las cuentas de usuarios. Los permisos de cada usuario deben ser determinados por las tareas que va a realizar con su rol en la organización.
2. La identificación de usuarios debe ser homogénea para todo el personal de la organización.
3. Se deben realizar verificaciones periódicas de los permisos que tiene cada usuario para confirmar que estos no han cambiado sin autorización por medio de auditorias o algún otro método.
4. Las revisiones deben orientarse a verificar que los permisos de acceso sean adecuados a sus necesidades operativas de cada usuario. Los permisos de acceso pueden ser redefinidos de acuerdo a las necesidades de la organización, además se pueden hacer análisis de las cuentas buscando periodos de inactividad o cualquier otro aspecto anormal que permita redefinir la necesidad de acceso.
5. Detectar actividades no autorizadas por medio de auditorias.
6. Mantener actualizados los tipos de permisos para determinadas tareas cuando se hace rotación de personal en la organización.
7. Procedimientos y políticas de acceso en caso de un despido o una renuncia de personal sea en forma amistosa o no, ya que por ejemplo los despidos del personal tienen un alto riesgo pues a veces estas personas tienen la capacidad de modificar el sistema instalando bombas lógicas , virus o troyanos que afectan al sistema o su información. O modificando los datos para que contengan información errónea o sin sentido.

Para evitar cualquier incidente es recomendable anular los permisos de acceso a las personas que por algún motivo se desvinculan de la organización lo mas pronto

posible, En caso de despido se sugiere que los permisos de acceso se anulen previamente a la notificación.

2.2 Puntos débiles de un sistema informático

Un intruso del sistema utilizará cualquier método, algoritmo o programa que haga más fácil su acceso y posterior ataque.

Existe una diversidad de frentes desde los que puede producirse un ataque. Esto dificulta el análisis de riesgos porque el delincuente aplica la filosofía de ataque hacia el punto más débil. De esta manera podemos ver diferentes tipos de ataque de la información, tales como la suplantación de identidad, la modificación de la información, o captura y posterior ataque del algoritmo de cifrado (Si es que acaso se ha utilizado uno)

Así un sistema es tan fuerte como el más débil de sus eslabones.

2.3 El tiempo que deberá protegerse un dato

Los datos confidenciales deben protegerse sólo hasta que ese secreto pierda su valor [ACA05]. Se habla, por tanto, de la caducidad del sistema de protección que es el tiempo en el que debe mantenerse la confidencialidad o secreto del dato. Esto nos llevará a la fortaleza del sistema de cifrado, por ello debemos hacer un análisis sobre el tiempo que debe mantenerse el secreto, de eso dependerá no solo el tipo de algoritmo que se debe utilizar para el cifrado de los datos, sino también a cuantos bits deberá cifrarse los datos

2.4 Medidas de control que se implementan para ser utilizadas de forma efectiva

La implementación del método que nos permita proteger la información no solo debe cumplir los estándares de seguridad básicos para tener un cifrado fuerte sino que además debe de cumplir algunos requisitos prácticos ya que sin estos se podría tener un excelente algoritmo de cifrado, pero ser un algoritmo completamente inviable de lo lento que procesa la información, o tal vez un algoritmo muy fuerte pero difícil de manejar por un usuario que no está familiarizado con un software especializado(ni le interesa estarlo, por ejemplo una secretaria o un poeta) [ACA05].

Por ello los algoritmos de cifrado implementados deben cubrir las siguientes condiciones:

- Deben ser eficientes, fáciles de usar y apropiados al medio.
- Que funcionen en el momento oportuno.
- Que lo hagan optimizando los recursos del sistema.
- Que pasen desapercibidos para el usuario.
- Y lo más importante: ningún sistema de control resulta efectivo hasta que es utilizado al surgir la necesidad de aplicarlo. Este es uno de los grandes problemas de la Seguridad Informática (Los planes de contingencia)

2.5 Amenazas de un sistema de información

En la implementación de seguridad en un sistema informático se pueden apreciar tres entes [BOR01]:

1. El poseedor del valor que es llamado Protector
2. Alguien que aspira a poseer el valor que es llamado Competidor-Agresor
3. Un elemento a proteger que es el Valor.

En donde la seguridad se define como: “La interrelación dinámica (competencia) entre el agresor y el protector para obtener (o conservar) el valor tratado, enmarcada por la situación global.”

Cabe hacer algunas aclaraciones, entre ellas las siguientes:

1. El protector no necesariamente es el poseedor del valor.
2. El agresor no siempre es el aspirante a ser poseedor
3. Ambas figuras pueden ser delegadas a terceras personas a cambio de otro valor, principalmente por, dinero.
4. El valor no necesariamente es algo concreto o algo físico. Por ejemplo se podría querer cuidar el conocimiento de algo, o el secreto de algo.

Además los competidores se pueden dividir en:

- Competidor interno: Aquel que piensa que la organización está por encima de sus intereses, y por lo tanto actúa para sobreponer su interés personal, provocando daños a la organización.
- Competidor externo: El que actúa para arrebatar al poseedor lo que para él significa un valor empresarial o personal (clientes, mercado, información, investigaciones tecnológicas, etc.)

La seguridad es un problema de antagonismo y competencia. Si no existe un competidor-amenaza el problema no es de seguridad.

Además según Cristian F. Borguello [BOR01] comenta que para hacer un análisis de Seguridad informática se deberán conocer las características de lo que se pretende proteger, en este caso la información.

Se define Dato como la unidad mínima con la que se compone cierta información, esta palabra proviene de Datum que en latín significa “lo que se da”.

Además se define información de la siguiente manera: Es una agregación de datos que tiene un significado específico más allá que cada uno de estos, y tendrá un sentido particular según como y quien lo procese.

Establecer el valor de la información es algo completamente relativo, ya que lo que para una organización es muy importante le es completamente indiferente a otra, lo mismo ocurre por ejemplo con investigaciones de diferentes áreas que no se conectan de ninguna manera una con la otra.

En la iniciativa privada o en el gobierno existe información que puede ser del dominio público y esta puede ser leída y reproducida por cualquier persona como los productos que vende o realiza una empresa o una fábrica, o los requisitos para la petición de determinado trámite en el caso del gobierno. Pero también existe el tipo de información que debe ser secreta o clasificada que debe ser revelada solo a personal autorizado, como por ejemplo los números de tarjeta de crédito de la base de datos de los clientes de una empresa, ó el tipo de armas y la cantidad de ellas en las fronteras del país en el caso de Ejército. En esta última es en la que se deben realizar las tareas de preservación de manera secreta tomando en cuenta que esta información cumple las siguientes características [ACA05]:

- Es crítica y por ello es indispensable garantizar la continuidad de la misma
- Es valiosa, la información es un activo con valor en sí misma
- Es sensitiva ya que debe ser conocida sólo por las personas que la procesan y por los usuarios de ésta; pero solamente por ellas.

Por ello es importante conocer el tipo de amenazas de las que se necesita proteger la información, es necesario saber que tipo de información es la que se debe mantener en secreto, con qué objetivo y cuáles son las posibles amenazas de las que puede ser objeto, los posibles ataques y los planes que se tienen pensados en contra de esos posibles ataques, así como planes de contingencia en caso de que un ataque sea certero.

2.5.1 Tipos de amenazas de un sistema de información.

Las amenazas afectan principalmente al Hardware, al Software y a los Datos. Éstas se deben a fenómenos de:

- Interrupción.
- Intercepción
- Modificación
- Generación

La interrupción se refiere a que una información transmitida de un lugar a otro llega incompleta o no llega debido a varios factores, puede ser que el sistema eléctrico falló antes de acabar de transmitir el mensaje, o antes de que se recibiera, puede deberse a un ataque de negación de servicio donde por medio de un software este mensaje es rastreado interceptado y semidestruido, o modificado. O que el mensaje sea interceptado y se trate de generar otro mensaje que intente sustituir el mensaje original.

2.5.1.1 Amenazas de interrupción

Cuando se interrumpe la transmisión de la información se corre el riesgo de que la información se dañe, pierda o deje de funcionar en un punto del sistema. (Por ejemplo si la información y transmisión de datos es del sistema operativo se corre el riesgo de que la máquina quede en estado zombi, no importando el sistema operativo que se use, ni la cantidad de máquinas que se utilicen como en el caso de un cluster, ni la cantidad de procesadores que tenga a menos que posea un método especial de recuperación de información o algún sistema de tolerancia a fallos). Hay que aclarar que su detección es inmediata.

Ejemplo de estos puntos es la destrucción de hardware, ya sea de manera deliberada o por falla del mismo, el borrado de programas y datos de manera descuidada por personal de la organización, o por un intruso, o por un usuario malicioso, o por fallos del sistema operativo ya sea por que tiene uno o varios bugs, o por que un programa malicioso modificó su comportamiento, o por su mala administración, aunque también un pequeño fallo del suministro eléctrico puede alterar la información de los datos y provocar un fallo en el sistema operativo, o en la base de datos.

2.5.1.2 Amenazas de intercepción.

Cuando la información se transmite por un medio que, por naturaleza es inseguro, como en el caso de la Internet, se corre el riesgo de que esta información sea interceptada ya sea de manera accidental (Que muy raramente ocurre) o por alguien que le interesa interceptar esta información, si esta información es interceptada por personas no autorizadas se corre el riesgo de que estas persona tengan acceso a información de la que no se tiene permiso a acceder. Y así dependiendo del tipo de información a la que se tiene acceso se pueden adquirir privilegios que no se deberían tener. La detección de esta amenaza es muy difícil de detectar si no se incluye una firma digital a la hora de generar la información que se transmite.

Otro problema en la interceptación son las posibles copias ilícitas de programas o información, tales como cursos, libros electrónicos o música. Se escucha en la línea de datos por medio de un sniffer o alguna otra herramienta que pueda capturar la información o trama de alguna red.

2.5.1.3 Amenazas de modificación

Al tener una intercepción se tiene un acceso no autorizado de la información, esto puede ser utilizado en beneficio de la persona o personas que hayan interceptado la información.

La detección de este tipo de amenaza puede ser fácil o difícil dependiendo de las circunstancias en que se dé el incidente.

Es muy común que se modifiquen los datos en beneficio de la persona que lo interceptó como se mencionó anteriormente, comúnmente un beneficio económico, también se puede tener una modificación de hardware por ejemplo para poder acceder a las comunicaciones de manera directa por medio de la instalación de algún dispositivo físico.

2.5.1.4 Amenazas de generación

Al modificarse la información se crean nuevos objetos en los archivos o en el caso de que sea modificado el sistema operativo; se agregan nuevos objetos en él, como ya se mencionó es difícil detectar este tipo de amenaza pues no se tiene un antecedente de una firma digital al momento de generarse la información.

Ejemplos de este tipo de amenaza son los delitos de falsificación de información, añadir transacciones de red como en el caso de fraudes bancarios, ó añadir registros a una base de datos, como cuando se tiene una lista de pedidos por Internet y se le agrega más mercancía de la que se pagó y ordenó en realidad.

2.5.2 Amenazas características

Las amenazas características se clasifican de acuerdo al tipo de elemento que afectan [BOR01]:

- Al hardware:

  Los elementos naturales y los desastres que provocan pueden resultar una amenaza a los equipos entre ellos el agua, el fuego, el viento, elementos nocivos que afectan el equipo si se exponen al polvo, al humo de los cigarrillos, a la comida, insectos, roedores y si se tiene una sobre carga o una baja intensidad también en la corriente eléctrica.

- Al software:

  Además del ataque al hardware, se tienen los borrados accidentales o intencionados, la electricidad estática, fallos de líneas de programa (bugs), bombas lógicas que se activan en determinadas fechas o al ejecutar determinado comando, el robo de software o información, así como copias ilegales.

- A los datos:

  Tiene los mismos puntos débiles que el software. Pero hay dos problemas añadidos: no tienen valor intrínseco pero sí su interpretación la cual puede ser muy valiosa es decir al archivo como tal no dice nada pero la información que contiene puede ser muy costosa e importante para alguien en especifico, por ejemplo el personal en nomina ó las entregas a los clientes del mes; y por otra parte, algunos datos pueden ser de carácter público lo cual no tiene la mayor importancia pero el problema es cuando los datos no deben ser públicos como en el caso de los números de cuentas bancarias.

2.6 Requisitos de un criptosistema

Las propiedades específicas que el algoritmo de cifrado en la implementación debe cumplir los siguientes puntos [RAM04]:

- Algoritmo de cifrado y descifrado rápido y fiable. Para un uso fácil y confiable y transparente al usuario.
- Posibilidad de transmitir archivos por una línea de datos, almacenarlos o transferirlos.
- No debe existir retardo debido al cifrado o descifrado, o al menos el retardo no debe ser muy grande.
- La seguridad del sistema deberá residir solamente en el secreto de una clave y no de las funciones de cifrado, pues si el algoritmo es roto será fácilmente leído el mensaje, un ejemplo de algoritmo que puede ser roto sin la necesidad de conocer la contraseña es el algoritmo de cifrado DES.
- La fortaleza del sistema se entenderá como la imposibilidad computacional (tiempo de cálculo en años que excede cualquier valor razonable) de romper el cifrado o encontrar la clave secreta a partir de otros datos de carácter público.

.

2.7 Tipos de atacantes

Según la fuente en el año 2005 CSI/FBI en una encuesta realizada en las siguientes empresas [GOR05]:

Fig. 2.1. Encuestados por sector industrial

En la figura 2.1 se muestran los porcentajes de los tipos de empresas encuestadas por el CSI/FBI (Computer Security Institute/ Federal Buroe of Investigation) en su articulo Computer Crime and Security Survey.

Es importante conocer los tipos de incidentes que se tiene en la industria en general respecto al mal uso de los equipos, por lo que en la encuesta se muestra el comportamiento de los últimos 5 años respecto al uso no autorizado de los equipos teniéndose los resultados que se muestran a continuación:

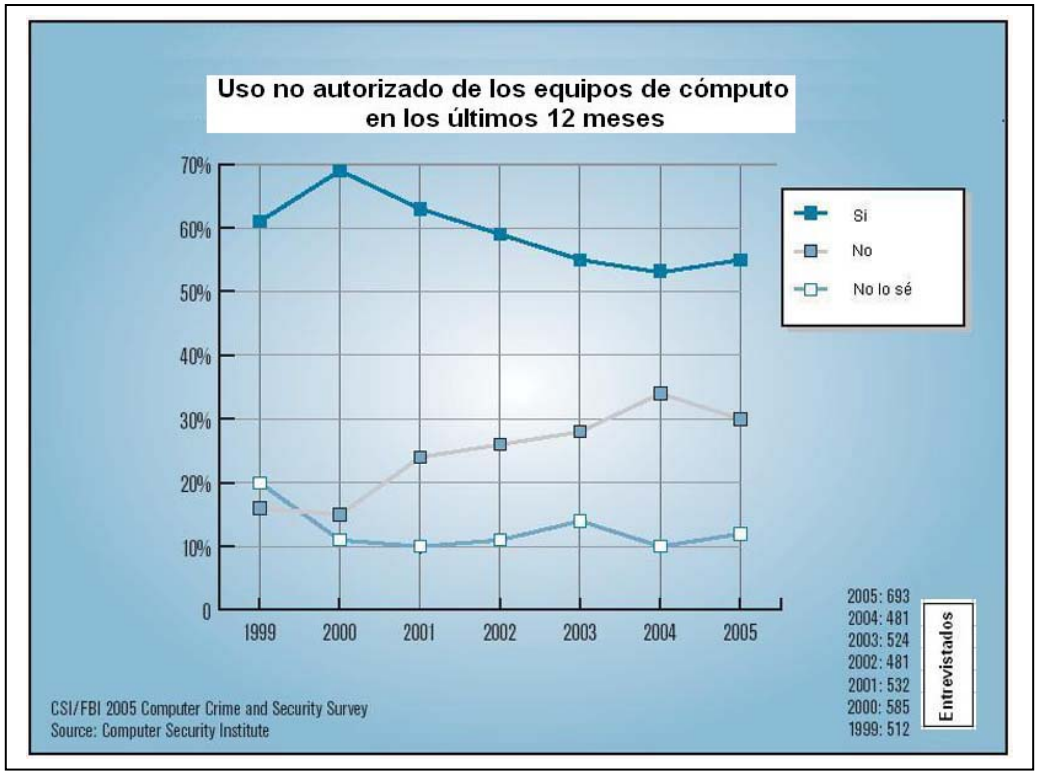


Fig. 2.2. Uso no autorizado de los equipos de cómputo en los últimos doce meses.

En la figura 2.2 se puede ver la frecuencia del éxito en el mal uso de los sistemas informáticos, este análisis comienza en el año 1999 y puede tener una baja este año. El porcentaje de encuestados que en su organización han experimentado un uso no autorizado en los últimos 12 meses ha incrementado ligeramente del 53% al 56% en 2005. Más aún, el porcentaje de encuestados que no han tenido un acceso no autorizado descendió de un 35 % a un 31 %. Los encuestados que no tenían idea si habían o no tenido accesos no autorizados se incremento de un 11 % a un 13 %

La información presentada en la tabla 2.1 indica que la frecuencia de los ataques se decrementa, el porcentaje de encuestados estima que sus compañías experimentan mas de 10 incidentes catalogados de bajo nivel para todas las categorías (Total de incidentes, incidentes originados desde fuera e incidentes originados desde dentro) desde 1988. Sin embargo hay un pequeño decremento en el árbol mayor a 10 incidentes del año pasado a este año que va del 12% al 9 % en los incidentes externos y del 8% al 3 % en los internos) Que puede ser debido al incremento de los que no conocen el dato por lo que la interpretación de la tabla es un poco problemática.

La tabla 2.1 sugiere que los encuestados detectaron mas eventos perpetrados desde dentro que desde afuera

¿Cuántos incidentes hay? ¿Cuántos internos? ¿Cuántos externos?

Cuantos incidentes por % de encuestados	1-5	6-10	>10	Desconocen el dato
2005	43	19	9	28
2004	47	20	12	22
2003	38	20	16	26
2002	42	20	15	23
2001	33	24	11	31
2000	33	23	13	31
1999	34	22	14	29
Cuantos incidentes desde afuera por % de encuestados	**1-5**	**6-10**	**>10**	**Desconocen el dato**
2005	47	10	8	35
2004	52	9	9	30
2003	46	10	13	31
2002	49	14	9	27
2001	41	14	7	39
2000	39	11	8	42
1999	43	8	9	39
Cuantos incidentes desde adentro por % de encuestados	**1-5**	**6-10**	**>10**	**Desconocen el dato**
2005	46	7	3	44
2004	52	6	8	34
2003	45	11	12	33
2002	42	13	9	35
2001	40	12	7	41
2000	38	16	9	37
1999	37	16	12	35

CSI/FBI 2005 Computer Crime and Security Survey
Source: Computer Security Institute

2005: 453 Encuestados

Tabla 2.1. Porcentajes de incidentes

Esto se debe principalmente a que las personas que trabajan dentro de una organización conocen mejor que nadie los movimientos que se realizan dentro de la misma, tipos de servidores que realizan determinadas tareas, nombres de dichos servidores, ubicaciones físicas, tipo de seguridad para tener acceso a dichos servidores ya sea de manera directa en consola o de manera remota, así como posibles vulnerabilidades que pudieran tener las aplicaciones instaladas en los servidores; todas estas cuestiones son conocimientos que debe tener un programador o un administrador del sistema informático de la organización para poder realizar su trabajo, de tal manera que el ataque de una persona con este perfil podría ser mas exitoso que los de personas que no tienen ningún tipo de conocimiento de la organización y parten de cero para realizar algún tipo de ataque.

Los motivos que llevan a una persona a realizar un ataque atentando contra la información y seguridad de una organización pueden variar que van desde chantajes, robos o factores que dependen de la conducta del individuo; lo importante es prever este tipo de ataques y estar preparados para que en la manera de lo posible sean evitados, o en caso de que estos sean exitosos, se tenga un plan de contingencia para reducir su daño.

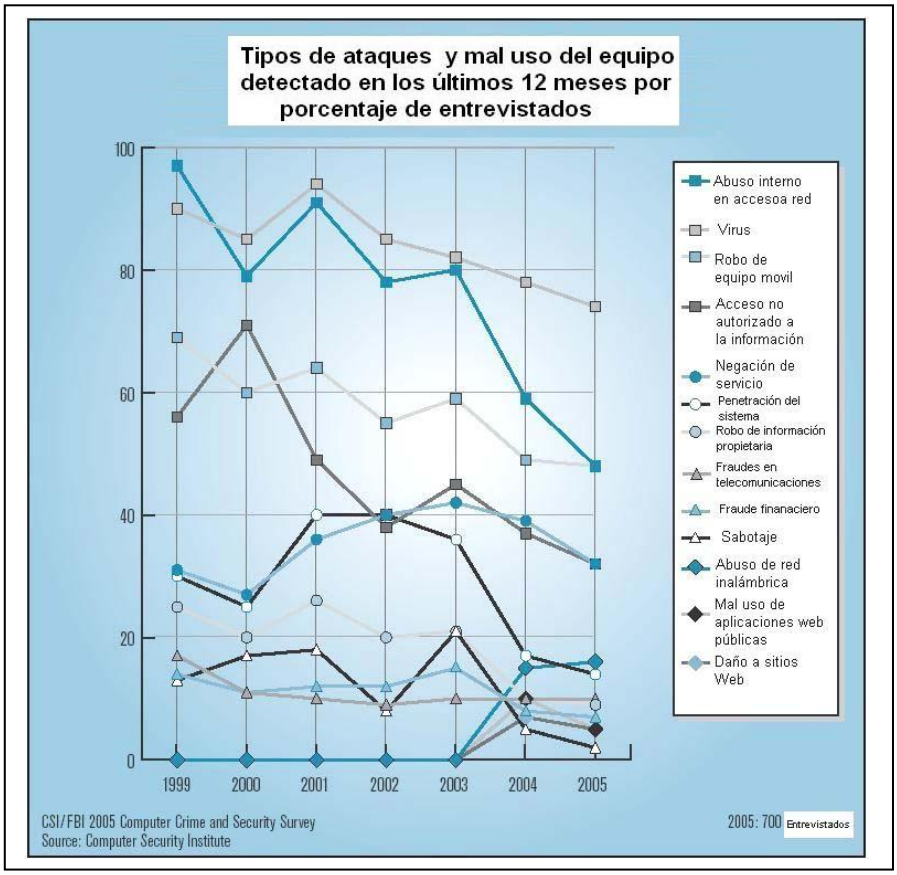


Figura 2.3. Tipos de atacantes y mal uso del equipo.

La figura 2.3 da una prueba visual estadística de los ataques detectados a los sistemas computacionales y del mal uso de ellos notándose que ha ido bajando el índice de incidencias con el paso de los años. Como se ve en la gráfica la única categoría que muestra un ligero aumento es la del abuso de redes inalámbricas, esta categoría junto con la de daños a sitios Web fue agregada desde el año pasado.

Figura 2.4. Pérdidas anuales por tipo de amenaza.

Encuestados estimaron que las pérdidas causadas por el tipo de incidentes respecto a años anteriores bajaron de manera dramática, las pérdidas totales en 2005 fueron de $130 104 542 para los 639 encuestados.

La figura 2.4 muestra las 3 categorías con más pérdidas que son los virus, los accesos no autorizados y el robo de información de la propiedad intelectual, además los daños a Web son los que menos pérdidas sufren. Es importante en base a esta estadística mantener cifrada la información ya que si el un servidor es penetrado y robada su información es importante que esta se siga manteniendo oculta por medio de algún método de cifrado.

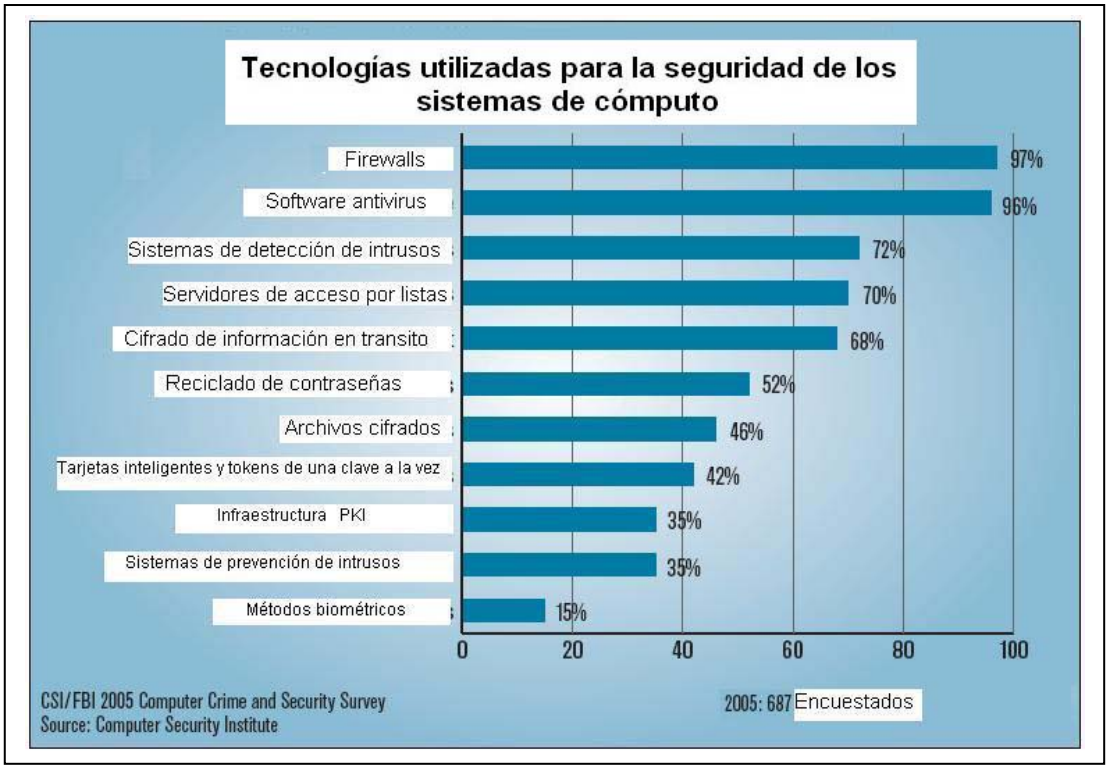


Figura 2.5. Tecnologías utilizadas para seguridad de los sistemas.

La figura 2.5 muestra los diferentes tipos de tecnologías ocupadas para la protección de los sistemas informáticos mostrándose una tendencia clara en el uso de firewalls seguido por el software antivirus y los sistemas de detección de intrusos y quedando en último lugar los sistemas biométricos de autenticación.

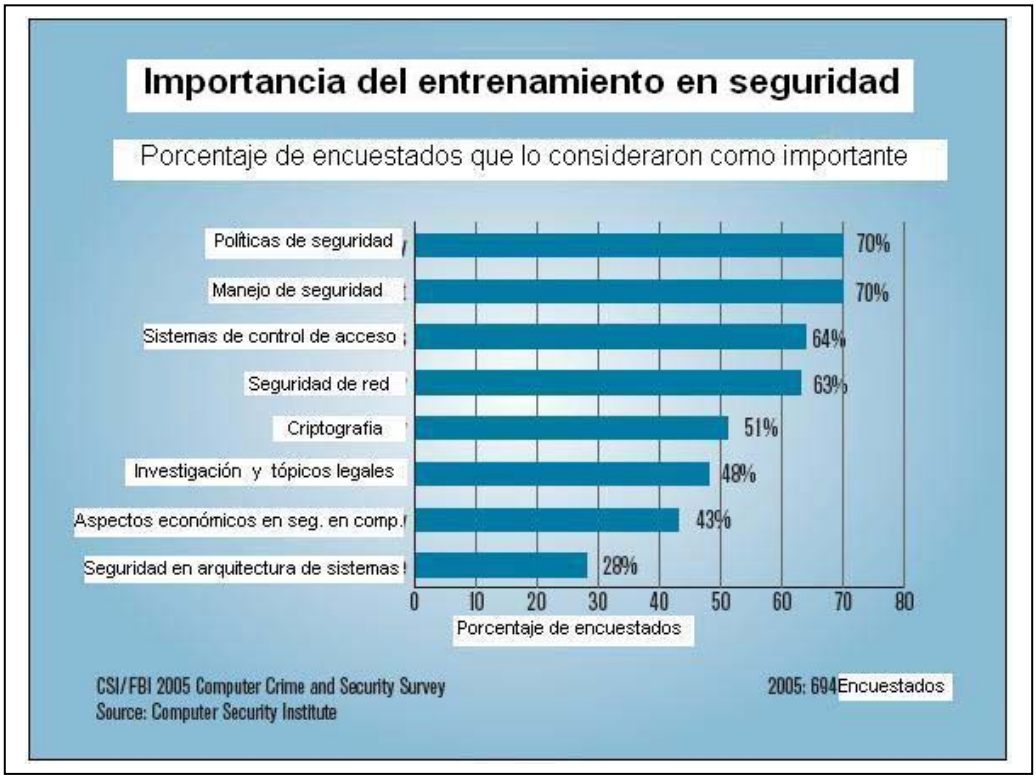


Figura 2.6. Importancia del entrenamiento en seguridad.

La figura 2.6 muestra los porcentajes de los encuestados sobre las tendencias de la importancia del entrenamiento en seguridad. Para 5 de las 8 áreas listadas en la tasa promedio indica que es muy importante el entrenamiento en las áreas correspondientes. Encabezando las 8 áreas se encuentran las políticas de seguridad con un 70% con el mismo porcentaje se encuentra el manejo de seguridad, seguidas por los sistemas de control de acceso con un 64%, la seguridad de la red tiene un 63%; hubo también cuatro áreas de seguridad fuertes identificadas en los últimos años (Aunque los porcentajes difieren al paso de los años). La quinta área de importancia es la de criptografía con 51%. El año pasado solamente el 28% de los encuestados identificaron esta área como la más importante para el entrenamiento.

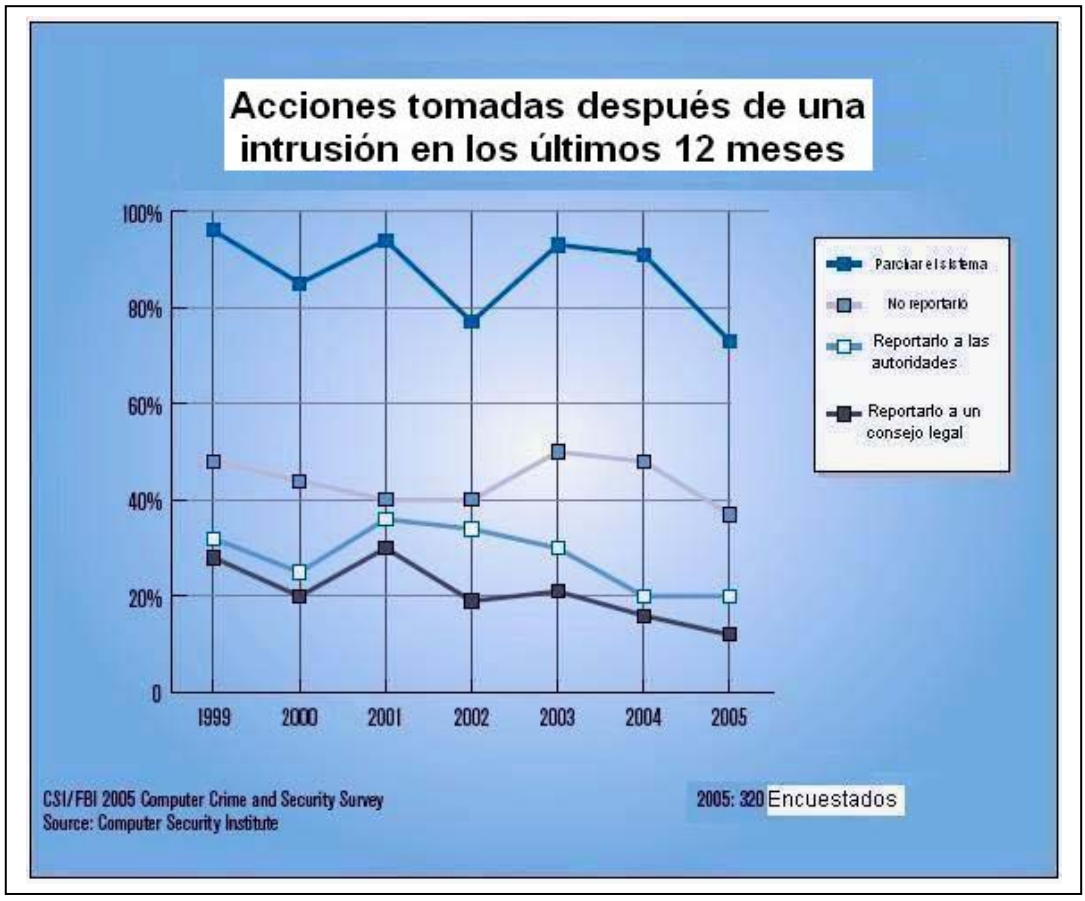


Figura 2.7. Acciones tomadas después de una intrusión.

En la figura 2.7 se muestra como las organizaciones responden a las intrusiones en cada año desde 1999. La gráfica indica que el 73 % de los encuestados indica que su empresa parcha los sistemas en los hoyos de seguridad. De manera sorprendente esta práctica ha descendido en este año y la tendencia es a la baja. Una posible explicación

puede ser que se han mejorado las automatizaciones en herramientas de parchado automático que hacen el proceso más transparente.

Figura 2.8. Razones por las que no se reportan los incidentes a la autoridad correspondiente.

En la figura 2.8 se dan las razones por las que las organizaciones no reportan las intrusiones a las autoridades. La razón predominante es la imagen de la compañía

Todas estas encuestas nos dan una fotografía del comportamiento de diversas empresas respecto a varios temas de interés en seguridad computacional y la manera en como tratan los problemas de intrusión, penetración de manera correctiva y preventiva en diferentes tipos de industrias.

De esta manera podemos darnos una idea de cómo se van dando las tendencias respecto a la protección de datos así como en los sistemas en los que se almacena la información.

2.7.1 Recomendaciones para evitar los ataques internos

Para reducir el riesgo de daño de un atacante interno se sugiere tomar las siguientes precauciones [SAN03]:

- Verificar los antecedentes de cada solicitante de empleo de la organización, aunque esto no garantiza que, si el prospecto no tiene antecedentes de haber hecho algo malo, con el tiempo no lo hará.
- Cada usuario debe saber sólo lo necesario para hacer su trabajo, es decir debe tener los conocimientos concisos y precisos para desempeñar su trabajo pero no más allá de sus funciones.
- Para evitar los problemas de complicidad en las fallas de seguridad se propone realizar rotación de funciones en la manera de lo posible para establecer una vigilancia mutua y así evitar las posibles complicidades.
- Se deben definir correctamente las funciones que desempeñara cada empleado de la compañía y no romper el esquema de funciones. Para que así no se viole la capacidad de conocer las funciones de otro personal ajeno a las funciones y obligaciones del primero. En concreto: Nadie tiene por que hacer las funciones que no le corresponden ya que con ello puede conocer secretos de la organización que le pueden abrir la puerta a información privilegiada.
- Una vez que un empleado ya no labora en la organización es muy importante cancelar sus cuentas en cada unas de las máquinas que empleaba, así como de su correo electrónico.

Estas medidas reducirán de manera importante las posibles fugas de información que pudiera haber por parte de ex empleados o de empleados corruptos.

2.8 Tipos de ataques

En particular para los datos se pueden tener varios tipos de ataques que se pueden ser los siguientes[BOR01]:

1. Ataques pasivos: El atacante no altera la comunicación, sino que solamente se dedica a “escuchar las comunicaciones” o monitorearlas, para obtener la información que esta siendo transmitida. Los principales objetivos son la intercepción de datos y el análisis de tráfico.

- Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitoreados.

- Control de volumen de tráfico intercambiado entre las entidades monitoreadas, obteniendo así información acerca de la actividad o inactividad inusual en los sistemas.
- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los periodos de actividad.

2. Ataques activos: Implican algún tipo de modificación del flujo de datos. Generalmente son realizados por crakers, piratas informáticos o intrusos remunerados y se les puede subdividir en cuatro categorías:

- Interrupción: si hace que un segmento de la información o toda ella quede inutilizable o no disponible
- Intercepción: Si un elemento no autorizado consigue el acceso a determinada información.
- Modificación: Si además de conseguir el acceso puede modificar la información.
- Fabricación: Se consigue un objeto similar al original atacando de forma que es difícil distinguirlos entre sí.
- Destrucción: Es la modificación que inutiliza la información.

2.8.1 Ataques al cifrado

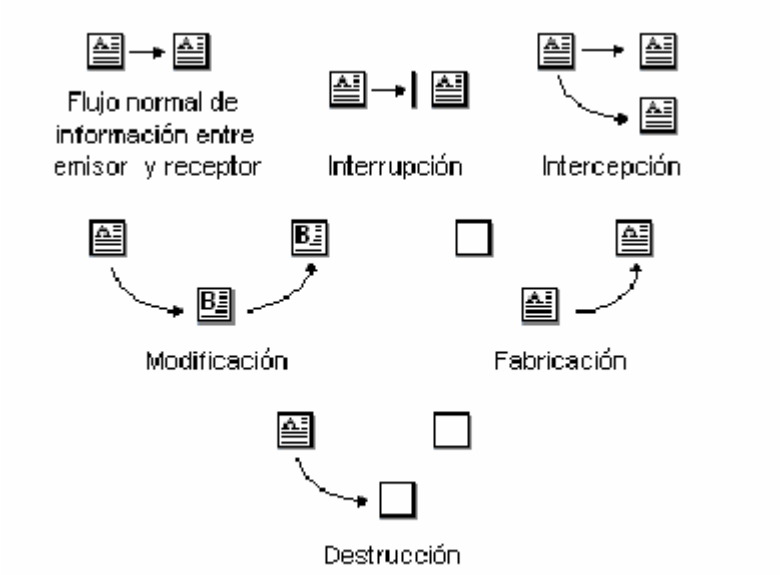


Fig. 2.9 Diferentes tipos de ataque

En la figura 2.9 se muestran los distintos posibles ataques que pudiera sufrir la información cuando esta viaja a través de un medio inseguro.

Conociendo el algoritmo de cifrado, el criptoanalista intentará romperlo por medio de alguno de los siguientes métodos [SAN03]:

1. Contando únicamente con el criptograma. Es decir por medio de algún método de espionaje se captura la comunicación cifrada y no se sabe nada del contenido del texto en claro.
2. Contando con texto en claro conocido. Se sabe el contenido del texto en claro y además se tiene este mismo texto cifrado mediante algún método de cifrado.
3. Eligiendo un texto en claro. Se elige un texto se cifra y se analiza que tan débil es el cifrado.
4. A partir de texto cifrado elegido. El texto se cifra con varias claves para ver cual es el comportamiento del algoritmo de cifrado y buscar posibles coincidencias dentro de las claves de cifrado.
5. Buscando combinaciones de claves (Ataque por fuerza bruta) Se hace un ataque del texto cifrado a partir de un generador de todas las claves posibles que pudiera tener para resolver la clave de cifrado.
6. A partir del algoritmo, puede ser que el algoritmo de cifrado sea bastante robusto y difícil de criptoanalizar pero el número de bits es muy bajo por lo que se presta hacer un ataque de fuerza bruta sobre la llave de bits que se genera al ejecutarse el algoritmo de cifrado sin tomar en cuenta la palabra clave de cifrado (contraseña)

2.9 Algunos ejemplos prácticos de sistemas criptográficos vencidos. [SAN03]

A continuación se muestran algunos ejemplos de algoritmos vencidos por diferentes medios y diferentes motivaciones, rompiéndose algunos mitos del cifrado y dando una gran lección no solo a las personas involucradas; sino también a cada una de las personas involucradas con datos que se deben proteger en las empresas.

2.9.1 La protección DVD que fallo (La seguridad por oscuridad no es efectiva)

Con la llegada del formato DVD se abrió un gran mercado para las películas con una gran resolución y una gran salida de audio, todo dentro de un disco compacto normal. Pero no todos se dieron cuenta de la lección, pues en cifrado gastaron una gran cantidad de dinero en torno a la DVD manía.

- No se debe de confiar en un algoritmo secreto o un algoritmo propietario aún si se trabaja para una institución gubernamental de seguridad nacional. El algoritmo será descubierto, solo es cuestión de tiempo, y si se conoce el algoritmo; se hace trivial el descifrado del mensaje si no se tiene una llave apropiada, y así todas las comunicaciones cifradas con este algoritmo son comprometidas.
- Nunca se debe confiar en una tecnología simple (o alguna otra medida) como la única línea de defensa. Se debe tener una defensa de varias capas con completéz y redundancia. Solo cifrar no es suficiente.
- Por sobre todo, nunca se debe escribir una técnica propia de cifrado. Pues ya existen algoritmos fuertes con implementaciones libres disponibles. A menos que sea un experimentado criptógrafo y se crea que el algoritmo es mejor que por ejemplo AES, blowfish, RSA ó RC4.

Pero ¿Qué fue lo que paso con el formato DVD? La industria del cine invirtió en años de investigación secreta desarrollando su propio estándar de seguridad, el sistema de contenidos intercambiados (CSS pos sus siglas en ingles). Este sistema permitió prevenir las duplicaciones de información no autorizadas de DVD por la contención de información cifrada en formato DVD. Cada DVD incluía una llave que podía ser utilizada para descifrar la información y una firma hash (Un valor de longitud fija formado a partir de texto plano) para verificar que la información no era incorrectamente descifrada. La llave era cifrada y podía ser descifrada solamente con un lector DVD, cuyas llaves estaban incluidas sobre el hardware de los lectores. En lugar de someter el estándar CSS a revisión, de lo que hubiera podido sacar ventaja de los criptoanalistas experimentados, ellos implementaron el estándar por cuenta propia y desarrollaron un producto que confiaba en sus sistema de cifrado.

De acuerdo con Frank Stevenson que publicó un criptoanálisis de CSS en noviembre de 1999, el cifrador fue diseñado con una llave a 40 bits (Que se indicaba en las especificaciones) para entrar en los estándares de exportación de los Estados Unidos

de América. Sin embargo solo se necesitaban 225 llaves en un ataque por fuerza bruta, El estimó que le tomaría menos de 18 segundos en una PC con procesador a 450 Mhz recobrar la llave del hash.

De acuerdo con Stevenson, “Si este cifrado era un intento de mantener la seguridad por oscuridad, entonces era una confirmación mas de que la seguridad por oscuridad es un principio inservible”

2.9.2 Tener cuidado con el exceso de confianza (Las llaves de gran longitud pueden no ser llaves)

En este caso se explora el riesgo del exceso de confianza en soluciones criptográficas. Todos los aspectos del criptosistema pueden ser sujetos de ataque, especialmente las llaves, A pesar de su importancia, las llaves nunca son protegidas de manera adecuada. Existen muchas situaciones que amenazan la integridad de las llaves. Una vez que una estación de trabajo es comprometida, capturar llaves es una tarea trivial. Una mala implementación del cifrado puede exponer las llaves temporalmente. Pero tal vez la causa más probable del compromiso de las llaves es la tendencia humana a fallar en la protección de llaves, almacenarlas en pedazos de papel debajo del teclado o dándole la contraseña por teléfono a quien asegura ser un administrador verificando la seguridad del sistema.

En 1998 Stephen Northcutt trabajó como analista técnico para ayudar a un equipo de agentes a detectar, investigar, atrapar y juzgar a un pornógrafo pederasta. Como dato interesante el pederasta utilizaba cifrado para la transmisión del material que no fue detectada por Northcutt por medio de un detector de intrusos (IDS), pues el detector no reconocía el patrón de una fotografía pornográfica debido a que estaba cifrada lo que no producía ninguna alerta.

¿Como lo atraparon? No fue difícil. La primera pista fue que se estaba transmitiendo una gran cantidad de información (Este tipo de transmisiones siempre causan sospechas), la siguiente pista es que aunque el IDS no detectaba el trafico cifrado este tenia una firma: “nariz blanca”. Se puede detectar que el tráfico esta cifrado solo con contar los bytes que son iguales, ya que los algoritmos cifran la información de tal manera que los bits de información sean muy parecidos para evitar ataques de similitudes de caracteres cifrados.

En este punto los agentes estaban listos para agarrar al sospechoso para un interrogatorio asumiendo que tal vez no sería posible descifrar la información a texto en claro. Pero examinando otras máquinas del pederasta se descubrió que la clave de acceso a la información estaba en un archivo en texto claro, por los que el sospechoso se fue a juicio.

Capítulo 3

3.1 Historia de las contraseñas

En las versiones más antiguas de MS Excel y Word, se guardaban las contraseñas en forma de texto plano o nativo (sin ningún tipo de cifrado) en la cabecera de los documentos protegidos [ISE04]. De esta manera, si se conseguía acceder a la cabecera del documento se podía leer la contraseña y violar el documento. Esto es válido para todas las versiones anteriores a Office 2000.

El sistema operativo Windows llegó a guardar las contraseñas con un formato de texto plano en un archivo oculto. En el caso de olvidar la contraseña se podía anular simplemente borrando el archivo oculto con lo que desaparecía la contraseña. A la fecha existen herramientas que pueden hacer la misma tarea, aunque el archivo ya no es texto plano, sino que está cifrado.

Pronto, Microsoft y Adobe empezaron a usar contraseñas, pero sólo para denotar que los documentos necesitaban de una clave de acceso para ser abiertos, pero no para leer la información.

Esto significaba que si el documento se abría con otra aplicación, como por ejemplo el bloc de notas (notebook) la contraseña no era necesaria y la información podía ser leída sin problemas.

Microsoft Access 2.0 podía ser abierto como un archivo de texto fácilmente, simplemente renombrando el fichero con extensión “.txt”. Haciendo esto se podía leer perfectamente la información contenida en la base de datos.

Los archivos de Adobe PDF 4.0 y anteriores se podían imprimir y, muchas veces, visualizar también usando lectores PDF de Linux o el Ghostview para Windows.

Las redes inalámbricas (wireless networks) tienen un problema con la encriptación, ya que la clave de encriptación se puede calcular una vez se ha capturado un elevado volumen de la información que se transmite por el aire [FLU04]. Actualmente, con la capacidad de cálculo que tienen las computadoras, cada vez se tarda menos en “crackear” las contraseñas.

La seguridad de los sistemas Bluetooth se considera muy fiable, una vez que el sistema está configurado. El problema es que Bluetooth transmite una contraseña única entre los dispositivos para establecer la conexión y ésta se envía como texto plano. Si esa contraseña es interceptada, toda transmisión futura durante esa sesión puede descifrarse fácilmente por un intruso.

De esta manera han surgido varias propuestas para proteger documentos e información y poco a poco todas ellas han sido vencidas, por lo que los protocolos y medidas de protección deben ir evolucionando conforme evoluciona el poder de cómputo y las aplicaciones en general.

3.2 La seguridad lógica

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para ello. [BOR01]

Los objetivos a plantear son:

- Restringir el acceso a los programas y archivos.
- Asegurar que los operadores puedan trabajar sin una supervisión minuciosa sin que puedan modificar los programas ni los archivos que no les correspondan.
- Asegurar que se estén utilizando los datos, archivos y programas correctos con el procedimiento correcto.
- Que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y no a otro.
- Que la información recibida sea la misma que la que se ha transmitido.
- Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos.
- Que se disponga de pasos alternativos de emergencia para la transmisión de información.

En el caso del control de acceso, este puede ser implementado a través del sistema operativo, sobre las aplicaciones que tratan la información, sobre las bases de datos, o mediante algún paquete específico para mantener la confidencialidad de la información. Es importante también restringir la cantidad de usuarios y procesos con acceso permitido y resguardar la información confidencial de accesos no autorizados.

3.3 Ataques por criptoanálisis

Cuando se desarrolla un algoritmo de cifrado, este da una visión desde entonces de los posibles tipos de ataques a emplear para romper el cifrado. Cada tipo de ataque debe tratar con piezas de información que un criptoanalista tiene. Se asume que un criptoanalista tiene un profundo conocimiento del algoritmo de cifrado. [SAN03]

Un criptoanalista con acceso al texto plano y al texto cifrado puede montar un ataque "conociendo el texto plano". En donde la meta es encontrar la clave empleada para cifrar o utilizar un algoritmo alterno para descifrar cualquier mensaje con una llave que el criptoanalista conozca.

Similar al ataque "conociendo el texto plano" esta el ataque del texto elegido. Para este ataque, el criptoanalista es capaz de elegir que texto plano toma para cifrar y analiza el resultado del texto cifrado. Y algunas veces es capaz de elegir correctamente que texto plano le puede dar algún indicio de la llave de las cadenas cifradas.

Un caso especial es el "ataque del texto plano adaptativo". Que después de elegir el texto plano que será descifrado. El criptoanalista puede elegir otros bloques para ser cifrados también. Este ataque permite aún más análisis basados en los resultados de cada paso del algoritmo.

NOTA: Todos los ataques anteriores requieren que el criptoanalista tenga el texto plano y el texto cifrado.

Los ataques "solo al texto cifrado" requieren solamente del mensaje cifrado, sin que el texto plano este disponible. El objetivo es recobrar uno o más mensajes o en el mejor de los casos la llave de cifrado utilizada.

En un ataque con conocimiento del texto, el criptoanalista puede elegir el texto cifrado a descifrar. Así el criptoanalista tiene un texto cifrado y un texto plano para elegir los mensajes, este ataque es usado principalmente contra cifrado de llave pública.

En un ataque con llave a elegir, el criptoanalista conoce alguna relación específica entre las llaves, al contrario de lo que el nombre sugiere, el criptoanalista no elige la llave sino más bien establece algunas relaciones respecto a la misma.

3.4 La seguridad de las contraseñas

Las contraseñas en todo tipo de documentos deben cumplir con una serie de propiedades que muchas veces no son obedecidas por los usurarios, los usuarios tienen por costumbre poner toda clase de contraseñas débiles. Se pone en claro las tácticas para poder romper una contraseña [BOT04] .

Casi en toda la Internet así como en los bancos en línea los usuarios permiten fraudes al utilizar contraseñas predecibles e ignoran los requerimientos elementales de la seguridad en cómputo.

La investigación [BOT04] muestra que el 21 % de las personas usan su nombre o el de su pareja como nombre base para sus contraseñas, el 15% usan sus fechas de cumpleaños o aniversarios de algún tipo y el 15% el nombre de su mascota. Alrededor del 14% el nombre de algún miembro de la familia, el 7% cofia en una fecha memorable, y el 2% carece completamente de imaginación pues usa como contraseña la palabra passwd. Solo una tercera parte de las personas admiten haber compartido la contraseña con su pareja, mientras que el 16 % se lo ha dicho a algún miembro de la familia y solo la mitad de estas personas sondeadas afirmo que absolutamente nadie conocía datos referentes a su clave o contraseña.

No es de sorprenderse que las personas malintencionadas ni siquiera deban usar sus destrezas de hacker para irrumpir en la computadora de alguien más. Sólo basta conocer detalles de su vida privada, ya que esto ayuda mucho. La forma más confiable de encontrar una contraseña es mediante la fuerza bruta o simplemente probando todas las palabras posibles que hay en un diccionario.

3.5 Tipos de contraseñas

En la sección 2.1.1 se habló de diferentes tipos de contraseñas a nivel teórico, ahora veremos en la práctica [SAN03] como que tipo de contraseñas hay y que se utiliza para realizar la autenticación de un usuario.

Existen principalmente tres tipos de contraseñas. Las que necesitan algo que sé, las que implican algo que tengo y las que añaden algo que soy.

3.5.1 Cadenas de caracteres

En el nivel más básico, las contraseñas son cadenas de caracteres, números y símbolos. Tener acceso a un teclado proporciona un método para introducir este tipo de contraseñas. Las contraseñas pueden ir de las más sencillas, como los tres números para acceder a ciertos estacionamientos o en algunas aulas y laboratorios de la U.N.A.M., hasta las más complicadas combinaciones de caracteres, números y símbolos que se recomienda emplear para proteger la información más sensible.

3.5.2 Cadenas de caracteres más un token

En el siguiente nivel, las claves requieren una cadena de caracteres, números y símbolos más un token o tarjeta de algún tipo. Un ejemplo típico es el de los cajeros automáticos. Para acceder a éstos se necesita una tarjeta y un número personal identificativo o PIN. Estos accesos se consideran más robustos ya que si se pierde u olvida alguno de los dos requerimientos el acceso será denegado.

3.5.3. Claves biométricas

El tercer nivel de complejidad son las claves biométricas. Consisten en utilizar alguna característica física no reproducible, como las huellas digitales o el aspecto de la cara, para permitir el acceso. Un ejemplo es el escáner de retina en el cual el interior del ojo se fotografía para la posterior identificación del sujeto. La retina contiene un patrón único de distribución de vasos sanguíneos fácilmente apreciable y que se puede utilizar para la identificación del individuo. Las contraseñas biométricas son las que se consideran más sofisticadas y seguras de todos los modelos de autenticación actuales.

Sin embargo, una clave de acceso que se pueda transportar en el dedo o en el ojo no tiene porqué ser más seguro que una transportada en la cabeza si el software está bien configurado.

3.6 Los errores más comunes a la hora de usar una contraseña

Es conveniente que nuestras contraseñas sean personales e importantes para nosotros mismos, pero también que sean difíciles de adivinar y que no se induzca tan fácilmente.

Survey Shop encuestó a 1,005 usuarios de la Internet por teléfono durante el mes de Marzo de 2004.

Además Hal Pomeranz comenta las desventajas comunes en el uso de contraseñas [POM03]:

- Se usa una contraseña como administrador y además de uso común para otras aplicaciones.
- Se usa una contraseña en más de una compañía por ejemplo con varios clientes si se es consultor.
- Se usa una contraseña como acceso a sitios de Internet, correo electrónico, Servidor IRC.

- Se usa una misma contraseña para dos arquitecturas diferentes.
- Se usa la misma contraseña en diferentes máquinas

El éxito del ataque por contraseña radica en que la mayoría de estas son reutilizables para cada aplicación que se utilice. Por ello, si un atacante es capaz de robar una contraseña, entonces esta puede ser usada hasta que dicha contraseña sea cambiada.

La cantidad de amenazas que existen contra las contraseñas va creciendo día con día, bastante tiempo, dinero y esfuerzo son empleados para tratar de encontrarlas, capturarlas y romperlas.

3.7 Maneras muy sencillas de robar contraseñas o inducir a poner contraseñas.

Hay cuatro maneras bien conocidas para robar contraseñas y se mencionan a continuación:

- Algunos atacantes intentan una búsqueda exhaustiva ya sea de manera manual o por medio de algún programa, por medio de la técnica de ensayo y error.
- Usuarios y administradores algunas veces escriben sus contraseñas en algún papel que pueden ser fácilmente robado por un atacante.
- Mirar por encima del hombro de alguien a la hora en que teclea su contraseña (O mirar por encima del hombro de alguien a la hora de que teclea su clave de acceso).
- Atrapar contraseñas en la red (Sniffing passwords) es muy común debido a que muchas redes usan recursos compartidos que prácticamente permiten que cualquier máquina conectada en el segmento tenga la posibilidad de observar el tráfico que pasa por el segmento. El usuario solo debe ejecutar un programa que haga esto, comúnmente llamado sniffer que mira en la red determinados blancos por medio de filtros, como por ejemplo la palabra “word:” capturando así las contraseñas u otra información sensible.

3.7.1 La ingeniería social.

Desafortunadamente es muy sencillo convencer a la gente de que hagan algo haciéndose pasar por alguna persona, institución o compañía en la que ellos confían. Para ejemplificar un típico caso de ingeniería social plantearé una situación hipotética:

Supongamos que el Señor X tiene información importante de la compañía. Y en una máquina a la que todos tienen acceso. Pero como es conciente de los peligros que

existen del robo de información él protege sus documentos de Microsoft Word por medio del método estándar que tiene este programa para hacerlo. Así el Señor X muy confiado en la seguridad que tiene para proteger sus documentos decide guardarlos con toda confianza en una carpeta con su nombre dentro de la carpeta de documentos compartidos.

El Señor K es una persona que ha visto como ha escalado exitosamente el Señor X en la empresa, desea perjudicarlo y no encuentra una manera convincente de hacerlo, de tal manera que el quede libre de culpa y pecado, su plan es el siguiente; él sabe que el Señor X tiene sus documentos a la vista de todos pero estos están protegidos por una contraseña, el Señor K ha intentado de todas las maneras que se le han ocurrido pero no ha podido encontrar la clave correcta, por lo que ha decidido hacer un ataque por ingeniería social.

Buscando en la red se ha enterado que algunas versiones antiguas de un servidor libre de correo puede ser vulnerado y así poder mandar un correo con una dirección y un usuario que el deseé, un correo que tal vez pueda convencer al Señor X que cambie las contraseñas de sus archivos.

El correo contendría algo como lo siguiente:

From: adminsitracion_seguridad@microsoft.com.mx
To: Señor_X@empresa_Y.com
Subject: Importante se ha encontrado una vulnerabilidad en nuestros productos.
Date: Mon, 25 Apr 2005 17:00:28 -0700
From: Sistemas y seguridad adminsitracion_seguridad@microsoft.com.mx

Por este medio hacemos de su conocimiento que nuestro programa Microsoft Word ha sido fuertemente vulnerado por medio de un ataque malicioso. Por lo que le pedimos de la manera mas atenta que si quiere conservar su información en secreto, usted deberá cambiar su contraseña con carácter de urgente a la palabra “3dplr5&34mzxt-34”
Para mayor información por favor consulte:

http://microsoft.com.mx

¡Por su atención y colaboración mil gracias!

Atentamente: Sistemas y Seguridad de Microsoft. Office

El Señor X recibe este correo y lo lee. ¿Qué hará el Señor X? El Señor X se ha tomado la molestia de leer algunos artículos relacionados a fraudes cibernéticos, por lo que tiene conocimiento de este tipo de fraude, por lo que no le hace el menor caso y además borra sus archivos del lugar de donde están a la vista de todo mundo.

Pero en este caso hipotético ¿Que es lo que podemos notar de este correo que tenga erróneo?

Por desgracia hay personas que caen en este tipo de ataque sin siquiera enterarse que han sido víctimas de una gran estafa. En primer lugar el correo es muy específico, ¿Cómo una compañía va a mandar a alguien un correo tan específico a alguien como el expuesto anteriormente? Si bien es cierto que Microsoft publica (a veces muy tardíamente) sus vulnerabilidades. No es costumbre de una compañía mandar correos a todo el mundo de sus vulnerabilidades a menos que se inscriba uno expresamente en una lista de seguridad de la compañía. Además este correo no contiene el número de vulnerabilidad o algo que identifique a esta fácilmente en la página oficial. Otro problema con el correo es el indicar que se debe cambiar la contraseña de los archivos protegidos, ¿Por qué alguien que es completamente un extraño va dictarme que contraseña poner en mis archivos? A menos que ese alguien quiera entrar a mis archivos sin mi autorización, aunque pareciera que la clave que me dictan aparenta ser muy segura en realidad, no lo es. Y no lo es precisamente por que alguien que no conozco me la esta dictando, por lo que desconozco en mano de quien o quienes esta esa contraseña y lo que puedan hacer con ella. Por lo que el sentido común nos dicta no hacer ningún caso de este tipo de contraseñas.

Es relativamente sencillo hacer este tipo de ataques y hay personas que caen es estos engaños pero por desgracia muchas veces es muy difícil detectar de donde vienen o por quien fueron creados, o ambas cosas.

3.8 Ataques por diccionario

En 1990 Dan klein describió algunas investigaciones realizadas por él en un muestreo de 14 000 cuentas en una variedad de sistemas. Creando varios diccionarios de palabras tanto comunes como palabras raras (términos relacionados con la ciencia ficción,

astronomía, gente famosa, referencias bíblicas, etc.) [POM03] El resultado fue sorprendente considerando la cantidad de contraseñas encontradas exitosamente usando un número relativamente pequeño de palabras y frases.

En particular la gente muestra una desconcertante tendencia a usar como contraseña su nombre de usuario, nombre de la maquina, su nombre propio (En este estudio en particular la palabra “Susan” fue la que mas coincidencias tuvo) o alguna palabra que se encontraba en el archivo /usr/dict/words. Un programa que ideó el autor de la investigación permutaba algunos caracteres comunes (Ceros en lugar de o, etc.) y muy a menudo se encontraron esta clase de contraseñas.

De hecho en la Tabla 3.1 se muestra que se requiere un pequeño número de palabras para hacer coincidir el 20% de las contraseñas recopiladas por Klein. En una computadora actual de escritorio se pueden revisar más de 50 000 contraseñas por segundo, mas adelante se verá mas detalladamente la cantidad de contraseñas que se deben revisar en base a la cantidad de caracteres por contraseña. Sólo se necesita acceder a una cuenta para poder romper la seguridad de una máquina.

Ataques por diccionario

Tipo de Palabra	**Tamaño de la búsqueda**	**Porcentaje acertado**
Nombres de usuario	130	2.7%
Nombre de la máquina	9018	1.0%
Nombres	7194	4.9%
Frases	933	1.8%
/usr/dict/words	19683	7.4%
Total:	36958	17.8%

Tabla 3.1

3.9 Ataques por Fuerza Bruta.

Un ataque por fuerza bruta a un mensaje cifrado para poder conocer su contenido en texto plano, consiste en descifrar un mensaje interceptado con cada posible llave y comparar el resultado con el texto plano conocido [COU04]. Aquí la dificultad radica en verificar cada una de las llaves posibles pues todo depende de este cálculo y

del tamaño del espacio a considerar, por lo que al ser un espacio pequeño de llaves encontrar la llave sea cuestión de segundos, así como encontrar un tamaño regular de llaves se lleve tal vez uno o dos meses, y un espacio de llaves grande se calcule en un par de años o en miles de años.

Por ejemplo si se tiene una llave a 40 bits el número total de diferentes posibilidades es de 2 a la potencia 40 la cual es 1,099,511,627,776 diferentes combinaciones binarias para cada una de las llaves. [SCI00]

3.10 Sugerencias para tener contraseñas fuertes.

A continuación se dan algunas recomendaciones para la generación de claves o contraseñas para su uso común [ETB05]. La contraseña permite la autenticación con nuestros equipos o aplicaciones en general para que pueda ser utilizado el servicio que se desea, si la contraseña no cumple con los requisitos mínimos puede ser fácilmente descifrada y utilizada para acceder a la información de manera no autorizada. Este caso como se ha visto en estadísticas anteriores se ha presentado en varias ocasiones (usuario: backup, contraseña: backup, usuario: abc, contraseña: 123, y otros ejemplos más).

El acceso a la información que maneja cada persona y la confidencialidad de la misma depende principalmente del mecanismo de autenticación, este se basa comúnmente en un usuario y una clave. Es por esto que es conveniente que tenga en cuenta las siguientes indicaciones para la elección de una clave adecuada:

- Debe utilizar al menos dos tipos de caracteres diferentes de los tres siguientes: letras (a-z, A-Z), números (0-9), caracteres especiales (#$%&/()=¡|...).
- Debe tener una longitud mínima de 8 caracteres.
- No se puede utilizar el mismo nombre de usuario como contraseña.
- No se deben utilizar nombres o apellidos, verbos o palabras que se encuentren en diccionarios, que faciliten deducir la contraseña.
- No usar nombres de series famosas como por ejemplo L0s_simps0n, simpson1 o cosas similares por el estilo.

Recomendaciones para la generación de contraseñas:

- No utilice información personal como fecha de cumpleaños, aniversarios, números telefónicos, números de identificación, profesión, nombre de las personas cercanas, etc.
- No utilice el nombre de usuario de correo, nombres o apellidos, combinados con números, con el fin de evitar que con un ataque de diccionario puedan lograr encontrar la clave que está usando.
- Se recomienda utilizar las iniciales de frases fáciles de recordar, combinando el uso de letras mayúsculas, minúsculas, números y caracteres especiales. Por ejemplo la frase "No todo lo que brilla es oro" podría quedar así: "Ntlqb30$", en este caso usamos las iniciales como las letras de la clave y cambiamos algunas letras por números similares: la letra E por 3 y o por 0.

Resumiendo una contraseña robusta es aquella que [ISE04]:

- No puede encontrarse en un diccionario
- Contiene números, letras y símbolos
- Contiene letras mayúsculas y minúsculas.
- Cuanto más largo, más robusto es.

Con una contraseña de 2 letras y 26 letras en el alfabeto, contando además con 10 números (ignorando los símbolos), hay 236 posibles combinaciones (687,000,000 posibilidades). Si aumentamos la longitud de la contraseña a 8 caracteres, ya disponemos de 836 combinaciones (324,000,000,000,000,000,000,000,000,000,000 posibilidades).

Hay muchos generadores de contraseñas robustas disponibles en Internet, pero éstos generarán una contraseña que es casi imposible de recordar.

Intente emplear, en cambio, una cadena aparentemente aleatoria de letras o números que se pueda recordar fácilmente.

Por ejemplo:

Ys=#1pt! (Yo soy el numero uno para ti)

ArJuAg1p (Ariadna, Juan Agustín y 1 perro – miembros de la familia)

LxRzDg24 (Alex Ruiz Diego – consonantes del nombre completo y la edad)

3.11 La ética al rescatar contraseñas, el Password Cracking (password Recovery)

El Password Cracking o el descifrado de contraseñas para propósitos ilegales es evidentemente ilegal [SAN03]. Pero si es su propia contraseña la que quiere descifrar, entonces estamos hablando de su información. Si de lo que se trata es de un individuo que está utilizando una contraseña para proteger algo, y entonces se olvida la clave de acceso, se necesita una recuperación de contraseña o password recovery.

El descubrimiento de contraseñas consiste en seguir unas técnicas básicas:

Echar una mirada alrededor: las contraseñas se guardan a menudo debajo de los teclados, bajo las alfombrillas del ratón o se cuelgan en las hojas “post-it” personales.

- La fuerza bruta: simplemente se prueban contraseñas de forma secuencial hasta que una funciona.
- Los ataques de diccionario automatizados: estos programas cruzan una serie de palabras pertenecientes a un diccionario hasta que una de éstas funcione como una contraseña válida.

Hay muchos programas disponibles en Internet que nos pueden ayudar con la recuperación de contraseñas introducidas en diferentes tipos de documentos. Sin embargo, cuanto más nueva es la versión del programa más fiable éste se vuelve y, por consiguiente, más difícil es obtener las contraseñas descifradas que usan, o encontrar un programa que nos ayude en la recuperación de la contraseña.

Capítulo 4

4.1 Matemáticas esenciales para el uso y entendimiento del cifrado [SAN03]

La criptografía es una especialidad matemática que incluye aspectos de teoría de probabilidad, teoría de la información, teoría de la complejidad, teoría de los números, álgebra abstracta y mucho mas. La propuesta en esta sección es dar una idea general de lo que es el cifrado en su aspecto matemático con algunas nociones matemáticas con las principales componentes que son utilizadas en general por los algoritmos de cifrado, ya sea por una sola técnica o por una combinación de ellas.

4.1.1 La substitución digital (El OR y el OR exclusivo)

El matemático George Boole a finales de 1800 inventó un algebra a la que llamó algebra lógica que cimentó las bases para las computadoras electrónicas y chips de microprocesadores. Sus operaciones lógicas fueron configuradas en tablas de verdad, en las cuales tanto las entradas como las salidas se definieron como los valores falso o verdadero.

La función booleana de OR exclusivo o XOR es una de las operaciones fundamentales utilizadas en los algoritmos de cifrado. La salida de XOR es verdadera si exactamente una de las entradas de la función binaria es verdadera, de otra manera la salida será falsa.

Por ejemplo:

- El cielo es azul (cierto) XOR La tierra es plana (falso) El resultado es cierto
- El cielo es azul (cierto) XOR La tierra es esférica (Cierto) El resultado es falso

En computación se requiere de números, por lo que se usan el 0 y el 1 en lugar de verdadero ó falso. La salida de una operación XOR se denota por el símbolo "$\oplus$", si se tiene que el resultado es cero, entonces las entradas son iguales y si se tiene que el resultado es 1, quiere decir que las entradas son diferentes.

Esta propiedad del XOR lo hace muy útil para el cifrado por dos razones. La primera es que al aplicar XOR a cualquier valor consigo mismo nos da como resultado $0 \oplus 0 = 0$, o bien $1 \oplus 1 = 0$; la otra utilidad de esta función es que el cero operado con algún valor, da este valor ya que $0 \oplus 0 = 0$, $0 \oplus 1 = 1$

¿Pero cuál es la utilidad?

Supongamos que Alicia tiene un mensaje que desea enviar en secreto a Bob y le envía la palabra en ingles “Buy”, esta se traduce en código ASCII de 7 bits en el flujo:

1000010 1110101 1111001

Ahora suponemos que Alicia y Bob ya tienen una llave secreta en común de 21 bits

0101101 1100111 1000101

Alicia convierte el texto plano en texto cifrado con la operación XOR por medio de su llave de la siguiente manera:

```
  1000010 1110101 1111001
⊕ 0101101 1100111 1000101
_____________________________

  1101111 0010010 0111100
```

Así Bob recibe el texto cifrado de Alicia y le aplica XOR con la misma llave secreta nuevamente para obtener el texto plano:

```
  1101111 0010010 0111100
⊕ 0101101 1100111 1000101
_____________________________

  1000010 1110101 1111001
```

De esta manera se obtiene nuevamente el texto plano que Alicia le mandó a Bob

Otra función booleana utilizada algunas veces en cifrado es la función OR, la salida de OR es verdadera si alguna de las entradas lo es; de otra manera la salida es falsa. Utilizando dígitos binarios, la salida es 1 si alguna de las entradas es 1; y la salida es 0 solo si ambas entradas es 0.

Así la función OR para las palabras anteriores será

1101111 0010010 0111100
0101101 1100111 1000101

1101111 1110111 1111101

4.1.2 La operación módulo

Muchas operaciones matemáticas del cifrado están cimentadas en el módulo aritmético. La función módulo es también llamada la función residuo. Es utilizada comúnmente en programación y es una operación básica en cifrado. Para calcular X modulo Y (expresado comúnmente como X mod Y) solamente se determina el residuo de Y dividido por X: EL valor de X mod Y estará ubicado en el rango 1 a Y-1

Ejemplos

15 mod 7 =1
33 mod 12 = 9
240 mod 200 = 40

4.1.3 El Método de sustitución

El objetivo principal del cifrado es transformar el texto de tal manera que sea inentendible. Los dos métodos básicos de cifrar o transformar el texto son la sustitución y la permutación. Un tercer método sería hacer una combinación de ambos. También existen dos métodos básicos de cifrado los de una llave (método simétrico) y el de dos llaves (método asimétrico). El algoritmo que se estudia en esta tesis es el algoritmo RC4 que es un algoritmo de una llave ó simétrico. Los algoritmos que utilizan dos llaves no serán vistos pues no es el objetivo de esta tesis.

Sustitución

La sustitución involucra el intercambio de un carácter (o byte) por otro. El esquema de sustitución simple utiliza un mapeo por lo que cada carácter será substituido por otro carácter para cifrar el mensaje, que al descifrarlo se hará la operación inversa. La función de mapeo es la llave, es decir alguien que conoce como se realizaron las sustituciones, puede realizar la operación inversa y descifrar el mensaje.

Ejemplo

Supongamos que definimos el siguiente mapeo con el abecedario

El texto plano es A B C D E F …
El texto sustituido podría ser: F W A C Q D …

Así al cifrar la palabra FEA se obtiene en su lugar la palabra DQF que es algo sin sentido en español. Por lo que la otra parte debe tener la regla de correspondencia para poder descifrar DQF por la palabra FEA. Es importante que la función de correspondencia sea uno a uno ya que de otra manera se puede malinterpretar el mensaje. Pues si por ejemplo si Q también sustituyera a T podría la palabra descifrada ser ambigua FTA que no tiene sentido en español.

4.1.4 El Método de sustitución por rotación

Otro método de substitución en donde el mapeo es más evidente es el método de substitución por rotación, en éste método se recorren las letras un determinado número de casillas, en este caso 7 lugares, de esta manera la A se transforma en G y la B en H y así de manera sistemática. También llamado el algoritmo César en honor a su inventor Julio César que de esta manera cifraba sus mensajes para mandárselos a sus generales. Este algoritmo rotaba 3 casillas y era considerado un buen algoritmo debido a que muy poca gente sabía leer y aún menos eran los que sabían escribir.

El texto plano es A B C D E F G…
El texto cifrado podría ser: G H I J K L M…

Así al cifrar la palabra FEA se obtiene en su lugar la palabra LKG que es algo sin sentido en español.

4.1.5 El Método de permutación

El método de permutación también es llamado método de transposición, cambia el orden en el que los caracteres (o bytes) aparecen más que substituir unos por otros.

Ejemplo

Supongamos que Alicia y Bob eligen la palabra clave SCUBA que sería 12345 para determinar el orden de permutación de caracteres, si aplicamos el método en la palabra

clave, se obtiene la palabra ABCSU, Pues la letra A es la primera letra, se le asigna la casilla número 1 y a U se le asigna la casilla número 5; así la cadena 43521 entonces determina la manera en que se moverán las letras. Alicia toma el mensaje, lo rompe en bloques de 5 caracteres (ya que es la longitud de la palabra) y entonces mueve los caracteres de acuerdo al bloque, si ella quiere enviar el mensaje:
DINNER TO NIGHT OK

Se parte en bloques de 5

D I N N E R T O N I G H T O K
1 2 3 4 5 1a 2a 3a 4a 5a 1b 2b 3b 4b 5b

Se recurre a la palabra clave y a su permutación 43521
Ella debe cifrarla en el siguiente texto:

N N E I D N O I T R O T K H G
4 3 5 2 1 4a 3a 5a 2a 1a 4b 3b5b 2b 1b

4.2 El algoritmo de cifrado RC4

En criptografía RC4 (ARCFOUR) es un cifrador simétrico de flujo diseñado por Ron Rivest de RSA Security en 1987 [WIK05]. Aunque en términos oficiales de RSA el término RC4 son las iniciales de “Rivest Cipher 4”, el acrónimo RC es comúnmente entendido como que es el código de Ron. También son del dominio público sus cifradores por bloque RC2 y RC5.

4.3 Cómo el algoritmo se hizo público

RC4 fue inicialmente un algoritmo comercializado como secreto, esto quiere decir que solo se tenía acceso al binario del algoritmo y no se sabía como estaba implementado ni su algoritmo teórico; pero en septiembre de 1994 una descripción de éste fue anónimamente publicada en la lista de correo Cypherpunk [GOL98], [WIK05]. Fue publicado rápidamente en el grupo de noticias de sci.crypt y de ahí a muchos otros sitios en la Internet. Como el algoritmo fue publicado este dejo de comercializarse como algoritmo secreto, pero el nombre RC4 es una marca registrada. Este algoritmo es frecuentemente llamado ARCFOUR para evitar problemas legales con la marca registrada, además se ha

convertido en parte común de algunos protocolos de cifrado y estándares utilizados frecuentemente, incluyendo WEP y WAP para tarjetas wireless y SSL.

4.4 Como se realiza teóricamente el cifrado [GRO00], [GOL98], [COU04]

RC4 es un cifrador de flujo, y como tal es básicamente un generador de números pseudo aleatorios inicializado desde una llave secreta por arriba de los 256 bytes. El algoritmo RC4, genera una clave de flujo la cual es simplemente pasada por la función XOR para producir el flujo cifrado. Para descifrar se utiliza exactamente la misma función de cifrado. Una de las razones a las que debe su popularidad es su sencillez. El algoritmo puede ser memorizado y rápidamente implementado, adicionalmente es ideal para implementaciones de software, pues requiere solamente manipulaciones de longitudes de un byte. Usa 256 bytes de memoria para los estados del arreglo, de S[0] pasa a S[255] , n bytes de memoria para la llave, de la Key [0] pasando a través de key[255], así como variables enteras i, j, k. n esta definida como el número de bytes en la llave y puede tener un rango de entre 1 < = n <= 255, pensando en aplicaciones comunes n =8 ó n =16 es común (64 y 128 bits, respectivamente), en este caso se describirá con n =5 que es una llave de 40 bits.

El algoritmo RC4 consta de dos partes, el primero comienza con la inicialización de la llave del algoritmo (KSA), que usa la llave para inicializar el número generador pseudo aleatorio:

```
-----------------------------------------------------------------------------------------------------
|  j = 0                                                                                            |
| for i = 0 .. 255                                                                                  |
|       S[i] = i                                                                                    |
| for i =0 .. 255                                                                                   |
|       j =  (j + S[i]) + key [i mod key_length] ) mod 256                                          |
-----------------------------------------------------------------------------------------------------
```

Una vez que se ha inicializado, ambos; cifrado y descifrado son ejecutados utilizando valores que salen a partir del estado de generación. Este proceso, llamado Algoritmo de Generación Pseudo Aleatoria (Pseudo-Random generation Algorithm PRGA) se desarrolla de la siguiente manera:

```
| i = 0                                                                                                  |
| j = 0                                                                                                  |
| ciclo hasta que el mensaje entero sea cifrado ó descifrado                                             |
| i = (i + 1) mod 256                                                                                    |
| j = (j + S[i]) mod 256                                                                                 |
| swap (S[i], S[j])                                                                                      |
| k = S [(S[i] + S[j] ) mod 256]                                                                         |
|  Salida del algoritmo XOR de k con el siguiente byte de entrada                                        |
```

Se recomienda de manera especial que la primera salida se descarte y no se utilice para cifrar mensajes (se recomienda descartar al menos 256 caracteres para mayor seguridad).

RC4 es un cifrador de flujo que utiliza el principio de cifrado simétrico ya que utiliza la misma clave para cifrar que para descifrar como lo muestra la imagen siguiente:

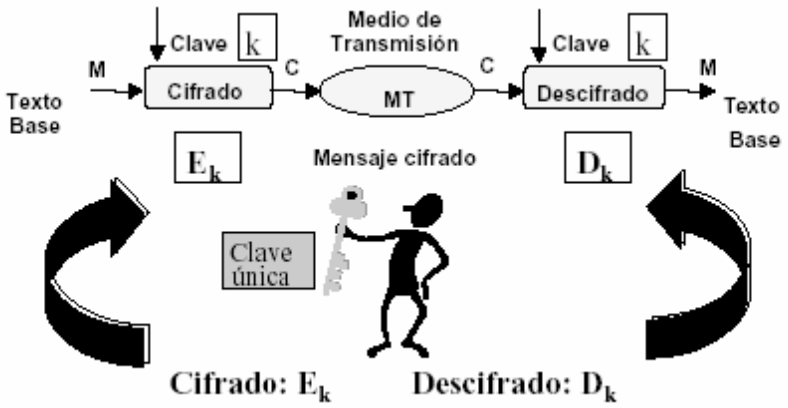


Figura 4.1

En la Figura 4.1 se muestra el uso de un algoritmo que usa la misma clave para cifrar o para descifrar llamados comúnmente los algoritmos de llave simétrica, como en el caso de RC4.

RC4 es uno de los cifradores más rápidos utilizados en aplicaciones serias.

Un criptoanálisis de RC4 tiene algunos estados inciertos. Cuando es utilizado de manera incorrecta, es posible romperlo de manera muy sencilla, si los bytes de texto plano son conocidos.

4.5 Ataque de Fluhrer, Mantin, y Shamir

En 2001 una nuevo y sorprendente descubrimiento fue hecho sobre todas las posibles llaves de RC4, las estadísticas para los primeros bytes son increíblemente no aleatorios, lo que da como resultado la posibilidad de descubrir la llave de RC4, las estadísticas para los primeros bits de la salida del flujo son en su mayoría no aleatorios, lo que da como resultado la posibilidad de descubrir la llave después de un gran número de mensajes cifrados con esta llave. Esto además de los efectos relacionados fueron usados para romper el WEP (wired equivalent privacy) cifrado utilizado con las redes inalámbricas 802.11, WEP emplea el algoritmo RC4 con llaves muy parecidas, abriendo la posibilidad de un ataque [FLU04]. Lo que causó un intercambio de estándares basados en el reemplazo de WEP en el mercado del 802.11, y se deja a la IEEE 802.11 la responsabilidad de la definición del protocolo a la WPA.

Una forma de evitar este problema es descartar los primeros o más de 256 bytes del cifrador de flujo.

Como cifrador de flujo es fácilmente rota, si la llave es utilizada dos veces por lo que se recomienda utilizar el valor hash de la llave.

4.6 El algoritmo en lenguaje C.

A continuación se presenta la publicación original en la que se presenta el código fuente o una propuesta de él, para poder realizar el cifrado [STE94]

```
Grupos  Viewing message <sternCvKL4B.Hyy@netcom.com>

  Free RSA encryption • Simple, Secure, Easy, Free to Use
Everyone should use - PC Magazine. •
download.cypherix.com/free_RSA Enlaces patrocinados

  Learn about cryptography • Caesar's cipher to AES. Securing
your comms in an insecure world • www.ftl.co.uk


Autor:David Sterndark (sterndark@netcom.com)
```

```
 Asunto:RC4 Algorithm revealed.
 View: Complete Thread (90 artículos)
 Original Format
 Grupos de noticias:sci.crypt, alt.security,
comp.security.misc, alt.privacy
 Fecha:1994-09-14 21:40:35 PST


 I am shocked,  shocked, I tell you,  shocked, to discover
 that the cypherpunks have illegaly and criminally revealed
 a crucial RSA trade secret and harmed the security of
 America by reverse engineering the RC4 algorithm and
 publishing it to the world.

 On Saturday morning an anonymous cypherpunk wrote:


    SUBJECT:  RC4 Source Code


    I've tested this.  It is compatible with the RC4 object
module
    that comes in the various RSA toolkits.

    /* rc4.h */
    typedef struct rc4_key
    {
         unsigned char state[256];
         unsigned char x;
         unsigned char y;
    } rc4_key;
    void prepare_key(unsigned char *key_data_ptr,int
key_data_len,
    rc4_key *key);
    void rc4(unsigned char *buffer_ptr,int buffer_len,rc4_key
* key);


    /*rc4.c */
    #include "rc4.h"
    static void swap_byte(unsigned char *a, unsigned char
*b);
    void prepare_key(unsigned char *key_data_ptr, int
key_data_len,
    rc4_key *key)
    {
         unsigned char swapByte;
```

```
          unsigned char index1;
          unsigned char index2;
          unsigned char* state;
          short counter;

          state = &key->state[0];
          for(counter = 0; counter < 256; counter++)
          state[counter] = counter;
          key->x = 0;
          key->y = 0;
          index1 = 0;
          index2 = 0;
          for(counter = 0; counter < 256; counter++)
          {
               index2 = (key_data_ptr[index1] + state[counter]
+
                  index2) % 256;
               swap_byte(&state[counter], &state[index2]);

               index1 = (index1 + 1) % key_data_len;
          }
     }

     void rc4(unsigned char *buffer_ptr, int buffer_len,
rc4_key *key)
     {
          unsigned char x;
          unsigned char y;
          unsigned char* state;
          unsigned char xorIndex;
          short counter;

          x = key->x;
          y = key->y;

          state = &key->state[0];
          for(counter = 0; counter < buffer_len; counter ++)
          {
               x = (x + 1) % 256;
               y = (state[x] + y) % 256;
               swap_byte(&state[x], &state[y]);

               xorIndex = (state[x] + state[y]) % 256;

               buffer_ptr[counter] ^= state[xorIndex];
           }
           key->x = x;
```

```
          key->y = y;
     }

     static void swap_byte(unsigned char *a, unsigned char
*b)
     {
         unsigned char swapByte;

         swapByte = *a;
         *a = *b;
         *b = swapByte;
     }
```

Another cypherpunk, this one not anonymous, tested the output from this algorithm against the output from official RC4 object code

```
    Date: Tue, 13 Sep 94 18:37:56 PDT
    From: ekr@eit.COM (Eric Rescorla)
    Message-Id: <9409140137.AA17743@eitech.eit.com>
    Subject: RC4 compatibility testing
    Cc: cypherpunks@toad.com

    One data point:

    I can't say anything about the internals of RC4 versus
the
    algorithm that Bill Sommerfeld is rightly calling
'Alleged RC4',
    since I don't know anything about RC4's internals.

    However, I do have a (legitimately acquired) copy of
BSAFE2 and
    so I'm able to compare the output of this algorithm to
the output
    of genuine RC4 as found in BSAFE. I chose a set of test
vectors
    and ran them through both algorithms. The algorithms
appear to
    give identical results, at least with these key/plaintext
pairs.

    I note that this is the algorithm _without_ Hal Finney's
    proposed modification
```

(see <199409130605.XAA24133@jobe.shell.portal.com>).

The vectors I used (together with the ciphertext they produce)
follow at the end of this message.

-Ekr

Disclaimer: This posting does not reflect the opinions of EIT.

```
--------------------results follow--------------
Test vector 0
Key: 0x01 0x23 0x45 0x67 0x89 0xab 0xcd 0xef
Input: 0x01 0x23 0x45 0x67 0x89 0xab 0xcd 0xef
0 Output: 0x75 0xb7 0x87 0x80 0x99 0xe0 0xc5 0x96

Test vector 1
Key: 0x01 0x23 0x45 0x67 0x89 0xab 0xcd 0xef
Input: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0 Output: 0x74 0x94 0xc2 0xe7 0x10 0x4b 0x08 0x79

Test vector 2
Key: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Input: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0 Output: 0xde 0x18 0x89 0x41 0xa3 0x37 0x5d 0x3a

Test vector 3
Key: 0xef 0x01 0x23 0x45
Input: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0 Output: 0xd6 0xa1 0x41 0xa7 0xec 0x3c 0x38 0xdf 0xbd
0x61

Test vector 4
Key: 0x01 0x23 0x45 0x67 0x89 0xab 0xcd 0xef
Input: 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
```

```
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
    0x01
    0 Output: 0x75 0x95 0xc3 0xe6 0x11 0x4a 0x09 0x78 0x0c
0x4a 0xd4
    0x52 0x33 0x8e 0x1f 0xfd 0x9a 0x1b 0xe9 0x49 0x8f
    0x81 0x3d 0x76 0x53 0x34 0x49 0xb6 0x77 0x8d 0xca
    0xd8 0xc7 0x8a 0x8d 0x2b 0xa9 0xac 0x66 0x08 0x5d
```

```
0x0e 0x53 0xd5 0x9c 0x26 0xc2 0xd1 0xc4 0x90 0xc1
0xeb 0xbe 0x0c 0xe6 0x6d 0x1b 0x6b 0x1b 0x13 0xb6
0xb9 0x19 0xb8 0x47 0xc2 0x5a 0x91 0x44 0x7a 0x95
0xe7 0x5e 0x4e 0xf1 0x67 0x79 0xcd 0xe8 0xbf 0x0a
0x95 0x85 0x0e 0x32 0xaf 0x96 0x89 0x44 0x4f 0xd3
0x77 0x10 0x8f 0x98 0xfd 0xcb 0xd4 0xe7 0x26 0x56
0x75 0x00 0x99 0x0b 0xcc 0x7e 0x0c 0xa3 0xc4 0xaa
0xa3 0x04 0xa3 0x87 0xd2 0x0f 0x3b 0x8f 0xbb 0xcd
0x42 0xa1 0xbd 0x31 0x1d 0x7a 0x43 0x03 0xdd 0xa5
0xab 0x07 0x88 0x96 0xae 0x80 0xc1 0x8b 0x0a 0xf6
0x6d 0xff 0x31 0x96 0x16 0xeb 0x78 0x4e 0x49 0x5a
0xd2 0xce 0x90 0xd7 0xf7 0x72 0xa8 0x17 0x47 0xb6
0x5f 0x62 0x09 0x3b 0x1e 0x0d 0xb9 0xe5 0xba 0x53
0x2f 0xaf 0xec 0x47 0x50 0x83 0x23 0xe6 0x71 0x32
0x7d 0xf9 0x44 0x44 0x32 0xcb 0x73 0x67 0xce 0xc8
0x2f 0x5d 0x44 0xc0 0xd0 0x0b 0x67 0xd6 0x50 0xa0
0x75 0xcd 0x4b 0x70 0xde 0xdd 0x77 0xeb 0x9b 0x10
0x23 0x1b 0x6b 0x5b 0x74 0x13 0x47 0x39 0x6d 0x62
0x89 0x74 0x21 0xd4 0x3d 0xf9 0xb4 0x2e 0x44 0x6e
0x35 0x8e 0x9c 0x11 0xa9 0xb2 0x18 0x4e 0xcb 0xef
0x0c 0xd8 0xe7 0xa8 0x77 0xef 0x96 0x8f 0x13 0x90
0xec 0x9b 0x3d 0x35 0xa5 0x58 0x5c 0xb0 0x09 0x29
0x0e 0x2f 0xcd 0xe7 0xb5 0xec 0x66 0xd9 0x08 0x4b
0xe4 0x40 0x55 0xa6 0x19 0xd9 0xdd 0x7f 0xc3 0x16
0x6f 0x94 0x87 0xf7 0xcb 0x27 0x29 0x12 0x42 0x64
0x45 0x99 0x85 0x14 0xc1 0x5d 0x53 0xa1 0x8c 0x86
0x4c 0xe3 0xa2 0xb7 0x55 0x57 0x93 0x98 0x81 0x26
0x52 0x0e 0xac 0xf2 0xe3 0x06 0x6e 0x23 0x0c 0x91
0xbe 0xe4 0xdd 0x53 0x04 0xf5 0xfd 0x04 0x05 0xb3
0x5b 0xd9 0x9c 0x73 0x13 0x5d 0x3d 0x9b 0xc3 0x35
0xee 0x04 0x9e 0xf6 0x9b 0x38 0x67 0xbf 0x2d 0x7b
0xd1 0xea 0xa5 0x95 0xd8 0xbf 0xc0 0x06 0x6f 0xf8
0xd3 0x15 0x09 0xeb 0x0c 0x6c 0xaa 0x00 0x6c 0x80
0x7a 0x62 0x3e 0xf8 0x4c 0x3d 0x33 0xc1 0x95 0xd2
0x3e 0xe3 0x20 0xc4 0x0d 0xe0 0x55 0x81 0x57 0xc8
0x22 0xd4 0xb8 0xc5 0x69 0xd8 0x49 0xae 0xd5 0x9d
0x4e 0x0f 0xd7 0xf3 0x79 0x58 0x6b 0x4b 0x7f 0xf6
0x84 0xed 0x6a 0x18 0x9f 0x74 0x86 0xd4 0x9b 0x9c
0x4b 0xad 0x9b 0xa2 0x4b 0x96 0xab 0xf9 0x24 0x37
0x2c 0x8a 0x8f 0xff 0xb1 0x0d 0x55 0x35 0x49 0x00
0xa7 0x7a 0x3d 0xb5 0xf2 0x05 0xe1 0xb9 0x9f 0xcd
0x86 0x60 0x86 0x3a 0x15 0x9a 0xd4 0xab 0xe4 0x0f
0xa4 0x89 0x34 0x16 0x3d 0xdd 0xe5 0x42 0xa6 0x58
0x55 0x40 0xfd 0x68 0x3c 0xbf 0xd8 0xc0 0x0f 0x12
0x12 0x9a 0x28 0x4d 0xea 0xcc 0x4c 0xde 0xfe 0x58
0xbe 0x71 0x37 0x54 0x1c 0x04 0x71 0x26 0xc8 0xd4
0x9e 0x27 0x55 0xab 0x18 0x1a 0xb7 0xe9 0x40 0xb0
```

```
    0xc0


 -----------------------------------------------------------
 We have the right to defend ourselves and our
 property, because of the kind of animals that we
James A. Donald
 are.  True law derives from this right, not from
 the arbitrary power of the omnipotent state.
jamesd@netcom.com
 -----------------------------------------------------------
```

4.7 Diferentes aplicaciones de RC4 en el mundo de uso generalizado

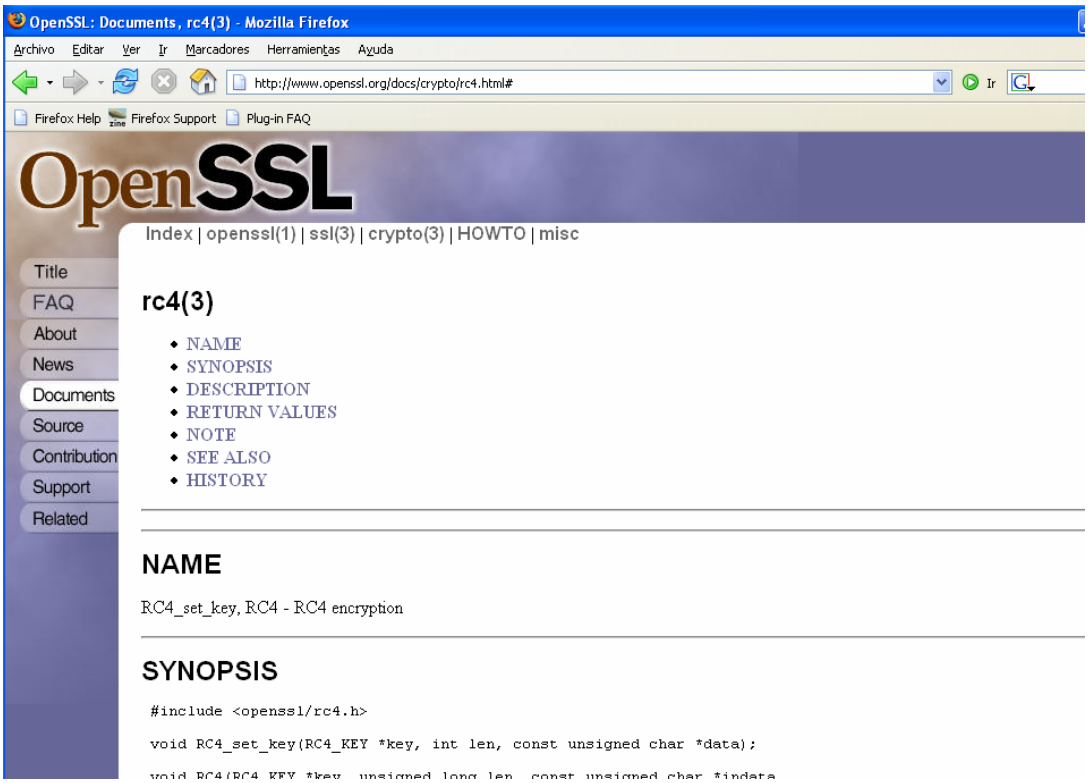


Figura 4.2

Openssl [COX05] contiene en sus rutinas el algoritmo RC4 para ser usado en una gran variedad de aplicaciones para el intercambio de datos, entre ellas el protocolo SSL/TLS.

El Protocolo SSL (Secure Socket Layer) fue desarrollado por Netscape Comunications en 1994 para proveer una aplicación independiente de comunicación segura en la Internet [SAN03]. Los procedimientos y rutinas de SSL son comúnmente empleadas en la Web por medio de protocolo de de transferencia de hipertexto (HTTP) para transacciones de comercio electrónico, además SSL no se limita solamente al protocolo HTTP. SSL se utiliza en criptografía para dar privacidad e integridad y autenticidad a los mensajes entre un cliente y un servidor.

La versión mas antigua de SSL que está en uso es la versión 2 (SSL v2), aunque la versión 3 (SSL v3) es la mas empleada, La Internet Engineering Task Force (IETF) se encarga de actualizar el SSL v3 pues crearon el protocolo no propietario de capa de transacción segura (TLS) descrita en el RFC 2246. En donde todo opera esencialmente de la misma manera. TLS está desarrollado como SSL v 3.1.
SSL/TSL emplea dos protocolos. El protocolo Handshake (Manejador) permite al cliente y servidor autenticarse mutuamente uno con el otro, intercambiando certificados y negociando los algoritmos de cifrado que serán utilizados. El protocolo de registro (record protocol) es utilizado para intercambiar la información cifrada.

SSL y TLS soportan varios protocolos de cifrado:

- SSL v2 soporta RC2 y RC4 con llaves de 40 bits para privacidad. SSL v4 soporta DES con llaves a 40 y 56 bits, RC4 con llaves de 128 bits así como 3 DES.
- Se utiliza para verificar la integridad del mensaje MD5 y SHA1
- Se utiliza RSA para el intercambio de llaves y de firmas digitales en SSL. Además TLS soporta para la misma tarea el algoritmo Diffie Hellman y el DSS.

El uso más popular de SSL y TLS es el asegurar que los usuarios de la Web no se están comunicando con un sitio pirata (spoofed server) o con un sitio comprometido, y asegurar que la comunicación sea entendible solo para los interesados en la comunicación aunque esta sea monitoreada por terceros. El servidor mantiene un certificado, firmado por una entidad certificadora por ejemplo Versión, Thawte o Banxico, el navegador se encarga de verificar que este certificado este en regla, de otra manera mandara un aviso de advertencia si es que detecta alguna anomalía, como en el caso de que el certificado haya sido modificado o haya expirado.

Cualquier solución de seguridad es tan fuerte como el mas débil de sus componentes, por lo que para la criptografía es importante diseñar e implementar con el mas alto grado de calidad posible. El ejército de los Estados Unidos de América utiliza criptografía desarrollada por la Agencia Nacional de Seguridad (NSA) para todas las comunicaciones clasificadas. La NSA no sólo proporciona Hardware, también provee llaves y políticas de uso. Y han desarrollado una infraestructura entera de criptosistemas pues ellos saben que es más importante proteger las comunicaciones en lugar de hacer algoritmos resistentes a los criptoanálisis.

Para el resto de la gente, que necesitamos o queremos hacer transacciones vía la Internet por medio del comercio electrónico, así como para las tiendas virtuales y universidades se dan otro tipo de soluciones. La criptografía provee un conjunto de herramientas que nos ayudan con la confidencialidad, integridad, autenticación y el no repudio. Esta herramienta (openssl) nos permite utilizar el protocolo https que es mas seguro que http, que es con el que hace uso de la banca electrónica. Con la cantidad de transacciones es muy difícil que alguien quiera capturar todas las operaciones que se hacen por ese medio, es más probable que un intruso más bien quiera dedicarse sólo a capturar una subred específica y trabajar con esa información. Pero para qué lidiar con el cifrado y descifrado, si se puede hacer una penetración exitosa en algún servidor en donde se almacenan los números de las tarjetas de crédito Este tipo de comunicación para transferencias bancarias utiliza ssl de softwares comerciales y puede utilizar en su lugar openssl , este tipo de aplicaciones no es un substituto para mantener un sistema seguro. Por motivos que ya se han explicado anteriormente.

Varios paquetes de cómputo utilizan open-ssl para el uso de sus rutinas de cifrado que incluyen el algoritmo RC4 a los bits que sea necesario. Entre ellas se encuentra openssh http://openssh.org así como stunnel http://stunell.org , entre muchas otras.

El paquete de Microsoft Office ampliamente utilizado a nivel mundial para una gran cantidad de personas de diferentes profesiones utiliza en su versión estándar para exportación, el algoritmo RC4 a 40-bits según las normas internacionales de cifrado. Para proteger los archivos de Word y Exel contra lectura pidiendo en cuanto abren la contraseña de acceso [GAT05].

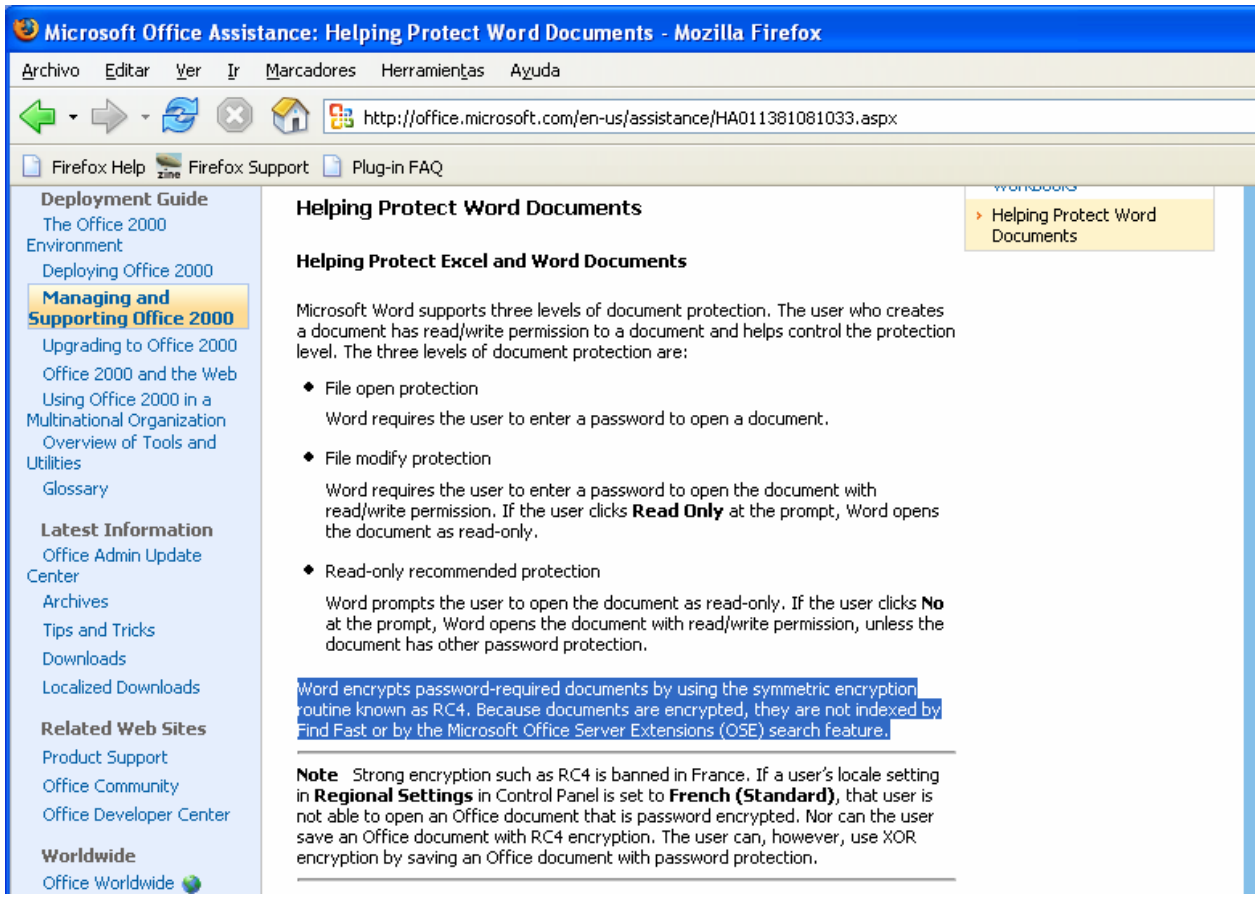

Microsoft Office Assistance: Helping Protect Word Documents - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://office.microsoft.com/en-us/assistance/HA011381081033.aspx
Firefox Help Firefox Support Plug-in FAQ
Deployment Guide
The Office 2000 Environment
Deploying Office 2000
Managing and Supporting Office 2000
Upgrading to Office 2000
Office 2000 and the Web
Using Office 2000 in a Multinational Organization
Overview of Tools and Utilities
Glossary
Latest Information
Office Admin Update Center
Archives
Tips and Tricks
Downloads
Localized Downloads
Related Web Sites
Product Support
Office Community
Office Developer Center
Worldwide
Office Worldwide
Helping Protect Word Documents
Helping Protect Word Documents
Helping Protect Excel and Word Documents
Microsoft Word supports three levels of document protection. The user who creates a document has read/write permission to a document and helps control the protection level. The three levels of document protection are:
File open protection
Word requires the user to enter a password to open a document.
File modify protection
Word requires the user to enter a password to open the document with read/write permission. If the user clicks Read Only at the prompt, Word opens the document as read-only.
Read-only recommended protection
Word prompts the user to open the document as read-only. If the user clicks No at the prompt, Word opens the document with read/write permission, unless the document has other password protection.
Word encrypts password-required documents by using the symmetric encryption routine known as RC4. Because documents are encrypted, they are not indexed by Find Fast or by the Microsoft Office Server Extensions (OSE) search feature.
Note Strong encryption such as RC4 is banned in France. If a user's locale setting in Regional Settings in Control Panel is set to French (Standard), that user is not able to open an Office document that is password encrypted. Nor can the user save an Office document with RC4 encryption. The user can, however, use XOR encryption by saving an Office document with password protection.

Figura 4.3

Capítulo 5

5.1 Criptoanalizando al algoritmo rc4.

En los cuatro capítulos anteriores se han visto una serie de características puntales que cubren los algoritmos de cifrado en general, que van desde la introducción a los algoritmos cifrados, así como la forma en que se debe proteger la información sensible ó valiosa; que es un activo más de la organización, pasando por los tipos de autenticación que hay para proteger el acceso información. En el capítulo anterior se trataron algunos conocimientos básicos de matemáticas para poder entender la estructura del algoritmo RC4, así como la forma en la que trabaja el algoritmo teórico, se describió quien lo hizo y la forma en que inicialmente salio a la venta, así como su posterior publicación anónima, lo que llevó a realizar código libre con este algoritmo, se mostró el código fuente publicado en la red en lenguaje C así como algunos de los protocolos y aplicaciones más importantes que utilizan este cifrado.

Lo que veremos a continuación es una descripción de los puntos que cumple el algoritmo en base a los 4 capítulos anteriores, describiendo las propiedades que cumple y por que las cumple lo que nos permitirá hacer una evaluación objetiva del algoritmo en si de manera teórica, posteriormente se realizará una propuesta de ataque al algoritmo a 40 bits por medio de ataques por diccionarios y ataques por fuerza bruta de hasta 8 cadenas de caracteres. En base a los resultados que nos arrojen las corridas para varios archivos en donde se tendrán criterios específicos para la selección de contraseñas se llegara a un análisis práctico de la fortaleza de las contraseñas del algoritmo a 40 bits.

5.2 Características teóricas

En primer lugar, como ya se mencionó, RC4 es un algoritmo que sirve para cifrar información, y la información se cifra para mantenerla libre del alcance de terceras personas, pues la información es un activo muy valioso para una organización.

El algoritmo fue hecho por R. Rivest, en este caso el algoritmo que cifra, es el mismo que descifra, que como se verá en los resultados del caso práctico se tiene un tiempo bastante razonable tanto para cifrar como para descifrar la información, el algoritmo es muy fiable, ya que hasta ahora, sólo se ha encontrado la vulnerabilidad del ataque de Fluhrer, Mantin, y Shamir por lo que para descifrar el algoritmo se debe encontrar la llave correcta y no se le puede aún dar la vuelta al algoritmo como en el caso de DES que sin saber la llave, por medio de un algoritmo específico se puede

obtener acceso a la información cifrada.

Como ya se vio, por medio de protocolos muy famosos se pueden transmitir archivos en una línea de transmisión insegura, pueden ser almacenados y pueden ser transferidos inclusive vía ftp.

Aunque un requisito del criptosistema es no tener retardo debido al cifrado y descifrado, por muy bueno que sea un algoritmo de cifrado cualquiera, se realizará un retardo en el cifrado o descifrado debido a varios factores intrínsecos de la arquitectura de una máquina, entre ellos la velocidad del procesador y la latencia de la comunicación entre los dispositivos, que en el caso de RC4 se encuentran en un parámetro razonable, además salvo las llaves encontradas vulnerables en el ataque de Fluhrer, Mantin, y Shamir, las demás llaves son seguras y fiables por lo que la seguridad reside solo en la llave y no en las funciones de cifrado (hasta ahora). Además en el caso práctico se verá que tanto tiempo computacional se requiere para romper el cifrado para encontrar la clave secreta.

En cuanto a las amenazas de diferentes tipos de personas ya sean internas o externas, el algoritmo se debe complementar con buenas políticas de seguridad, de acceso ya sea físico o remoto y mantener copias de seguridad de los datos, ya que el cifrado debe ser considerado como sólo una capa mas en la seguridad y no como un todo, ya que la seguridad es un proceso, no un producto y nuestro sistema de información será tan fuerte como el mas débil de sus eslabones.

RC4 es un cifrador de flujo, esto es que no tiene que dividir el mensaje y lo cifra completo, en este caso el espacio de claves es muy grande pues se puede poner prácticamente cualquier cadena finita de bits como clave para poder cifrar un mensaje, además no se puede reconocer de manera clara o fácil un patrón de cifrado ya que la llave es un flujo que va cambiando constantemente por lo que se tiene un periodo alto de secuencia de las cadenas cifradas, además cumple con las especificaciones de un cifrador genérico de llave automática dadas las especificaciones del la sección 4.4 aunque esto depende del número de bits en el que se cifra.

Con este algoritmo se pueden obtener satisfactoriamente la confidencialidad, la integridad de datos, la autenticación y el no rechazo combinándolo con otros algoritmos de la Criptografía. Como se muestra en la figura 5.1

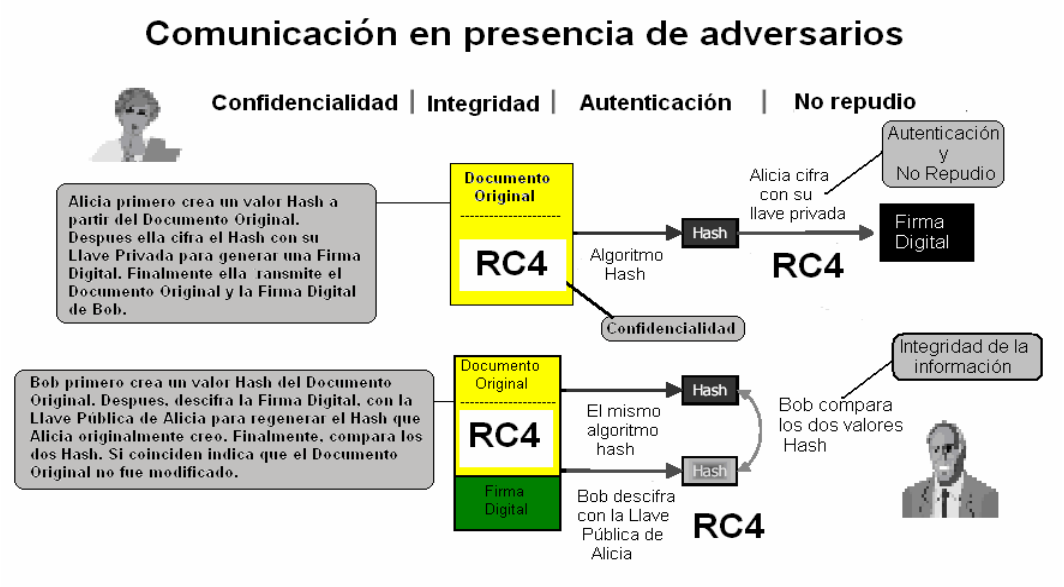


Figura 5.1 Es prácticamente la misma imagen de la figura 1.7 solo que aquí se esta empleando el algoritmo RC4 para realizar el cifrado de los mensajes. Por lo que el algoritmo cumple con los objetivos de seguridad para un algoritmo de cifrado.

Finalmente se debe tener en cuenta que tipo de claves se van a introducir a la hora de hacer un cifrado, por que existen diversas técnicas para obtener una clave, parece increíble pero me ha tocado ver de manera personal como máquinas tienen la contraseña de administrador o root tienen las claves pegadas en el gabinete. Y ya se explicaron varias maneras en las que se puede obtener de manera sencilla una contraseña. Cuando nada de esto funciona, se recurre a los ataques por diccionario y a los ataques por fuerza bruta que con tiempo y con un poco de suerte se obtendrá la clave deseada.

5.3 Características prácticas

En esta sección se describirá el tipo de ataques se realizarán con el algoritmo RC4 así como las características técnicas del mismo, se tomarán tiempos de proceso así como cantidad de operaciones por segundo dependiendo del tipo de ataque, así como el establecimiento de criterios para el cifrado de un archivo de texto y su posterior ataque.

5.4 Nuestro tipo de ataque

Nuestro ataque se realizará a partir de programas hechos en ANSI C con llamadas directas a las bibliotecas de Openssl (openssl/rc4.h) a continuación se muestran los programas compilados y corridos:

5.4.1 Programa cifra40.c

Este programa se llama `cifra40`.c el cual realiza el cifrado con el algoritmo RC4 a 40 bits de un texto cualquiera contenido en un archivo de texto y lo introduce a un archivo llamado `cifr` el programa toma de la línea de comandos el archivo a cifrar así como la palabra clave con la que será cifrado como se muestra a continuación:

$ cifra40 archivo.txt clave_de_cifrado
El código en ANSI C es el siguiente:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rc4.h>

//cifra40.c  29 de Octubre 2005
// Este archivo cifra un archivo cualquiera con una clave
dada
int main(int argc, char *argv[])
{
      //De leerachivo en var
      char nombre_archivo[50];
      char *sArchivote;
      long lTamano;
      RC4_KEY key;  //se define la varialbe key
      unsigned char data[16],buf[1024],*out,*out2;
      int outlen;
      //De leerachivo en var, para abrir el archivo a
cifrar

     FILE    *dfp;           /* archivo a cifrar */
```

```
     FILE *Archivo;          /* archive cifr en donde se va
a depositar */

     if (argc<3) {
        printf ("Use %s archivo llave\n",argv[0]);
        exit(0);
     }


     if ((dfp = fopen(argv[1],"r")) == NULL ) {
        perror (argv[1]);
        exit(-1);
     }

     memset(buf,'\0',sizeof(buf));
     strcpy (buf,argv[2]);

/*Si no son 2 argumentos (archivo y programa no se ejecuta
el archivo) y si no se abre el archivo a cifrar no se
ejecuta el programa */

     {


         // de leerachivo en var
               fseek (dfp,0L,SEEK_END);
               lTamano = ftell (dfp);
               sArchivote = malloc (lTamano);
               fseek (dfp,0L,SEEK_SET);
               fread (sArchivote,1,lTamano,dfp);
              fclose (dfp);
         // de leerarchivo en var


         RC4_set_key(&key,40,buf); //se inicai la llave
con clave de buf
          RC4(&key,  lTamano ,sArchivote,sArchivote);
```

```
         printf("Encrypted el sArchivote:\n----------
\n%s\n\n",sArchivote);
         Archivo = fopen("cifr", "w");

          fwrite (sArchivote,1,lTamano,Archivo);
         //Se escribe la variable en el archive cifr


          RC4_set_key(&key,40,buf);
         RC4(&key,  lTamano ,sArchivote,sArchivote);

          printf("Decrypted sArchivote:\n----------
\n%s\n\n",sArchivote);


           // de leer archivo en var
          free (sArchivote);
            // de leer archivo en var


       exit(1);

     }
}
```

5.4.2 Programa por_diccionario.c

En este programa se realiza un ataque por diccionario en donde se introducen los siguientes parámetros:

- El archivo a atacar
- El archivo en claro a comparar
- El diccionario que se va a utilizar en el ataque.

La manera en la que actúa éste programa es la siguiente:
$ por_diccionario cifr archivo.txt diccionario

El archivo cifrado es descifrado para cada clave tomada del archivo de palabras del diccionario comparándolo con el archivo de texto en claro cada vez para cada clave, si lo encuentra muestra la clave que sirvió para descifrar el archivo, de otra manera no muestra nada.

El código en ANSI C es el siguiente

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rc4.h>

//por_diccinario.c 30 de octubre  2005
//Se hace un ataque por diccionario en donde se dan las
contraseñas a revisar
//Se da el archivo cifrado a revisar así como el
diccionario a utilizar

int main(int argc, char *argv[])
{

// Del diccionario
char palabra[80], nombre[25];
char *c;
// del diccionario

//Del archivo a cifrar
 char nombre_archivo[50];
 char *sArchivote;
 char *sArchivote2;
 long lTamano;
//De archivo a cifrar

// Del archivo en texto plano a comparar con el cifrado
  char nombre_archivo1[50];
  char *sArchivote1;
  long lTamano1;
```

```
  FILE    *dfp;           //Archivo a analizar
  FILE    *dfp1;          //Archivo a compara en texto
plano
  FILE    *dfp2;          //Archivo de diccionario

  RC4_KEY key;  //se define la variable key

  if (argc<4) {
    printf ("Use %s cifrado claro diccionario\n",argv[0]);
    exit(-1);
  }
  if ((dfp = fopen(argv[1],"r")) == NULL) {
    perror (argv[1]);
    exit(-1);
  }
  if ((dfp1 = fopen(argv[2],"r")) == NULL) {
    perror (argv[2]);
    exit(-1);
  }
  if ((dfp2 = fopen(argv[3],"r")) == NULL) {
    perror (argv[3]);
    exit(-1);
  }

 {


 // de leerachivo en var
    fseek (dfp,0L,SEEK_END);
    lTamano = ftell (dfp);
    sArchivote = malloc (lTamano);
    sArchivote2 = malloc (lTamano);
    fseek (dfp,0L,SEEK_SET);
    fread (sArchivote,1,lTamano,dfp);
    fclose (dfp);
// de leerarchivo en var/

// del segundo archivo //
```

```
    fseek (dfp1,0L,SEEK_END);
    lTamano1 = ftell (dfp1);
    sArchivote1 = malloc (lTamano1);
    fseek (dfp1,0L,SEEK_SET);
    fread (sArchivote1,1,lTamano1,dfp1);
    fclose (dfp1);
   // del segundo archivo

      int cont = 0;   //CONTADOR

   printf ("Inicia\n");
  do
    {

     memset (palabra,0,sizeof(palabra));

     c = fgets(palabra, 41, dfp2);   /* Obtiene una linea
del archivo */
     if (palabra[strlen(palabra)-1]=='\n')  /*quita salto
de línea */
         palabra[strlen(palabra)-1]=0;

      printf ("Llave : %s\n",palabra);

     RC4_set_key(&key,40,palabra);  /* Se cifra y compara*/
     RC4(&key,  lTamano ,sArchivote,sArchivote2);
     cont = cont++;


     if (memcmp(sArchivote2,sArchivote1,lTamano)==0)
      {
        printf("AQUI SI LLEGA %d  \n", cont);
        printf("La clave es: %s \n", palabra);
        puts (sArchivote2);
        return -1;
      }
    }
     while (c != NULL);                  /* Hasta encontrar
```

```
NULL */

     fclose(dfp2);

 }
}
```

5.4.3 Programa fuerza_bruta_a_40.c

Este programa se llama `fuerza_bruta_a_40.c` y realiza una búsqueda exhaustiva de las posibles llaves a 40 bits, es decir un ataque por fuerza bruta sobre la contraseña

Ejecutándose de la siguiente manera:

```
$ fuerza_bruta_a_40 cifr archivo.txt
```

El programa busca todas las posibles combinaciones de llaves generándolas en la ejecución hasta completar los 40 bits es decir hasta 5 caracteres, ésta búsqueda incluye caracteres no imprimibles. Con cada clave el programa aplica el algoritmo al archivo cifrado y lo compara cada vez con el archivo en claro para cada clave hasta encontrar la clave correcta. Por medio de este método se obtiene la clave con un 100 % de certeza ya que se agotan todas las posibilidades para 40 bits que son el equivalente a 5 caracteres en código ASCII.

El código en ANSI C es el siguiente:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/rc4.h>

//ataque.c 30 Octubre 2005
//Se hace un ataque por fuerza bruta
//Se da el archivo cifrado a revisar así como el archivo a
comparar

int rc4_test (const char *sArchivote, const char
*sArchivote1, int lTamano, char *palabra, int nivel);
```

```
int main(int argc, char *argv[])
{

// Del archivo a comparar
char palabra[80], nombre[25];
char *c;
// del archivo a comparar

//Del archivo a cifrar
 char nombre_archivo[50];
 char *sArchivote;
 char *sArchivote2;
 long lTamano;
//De archivo a cifrar

// Del archivo en texto base a comparar con el cifrado
  char nombre_archivo1[50];
  char *sArchivote1;
  long lTamano1;

  FILE    *dfp;           //Archivo a analizar
  FILE     *dfp1;             //Archivo a compara en texto
plano
  FILE    *dfp2;

 /*   Se   valida   el   programa   con   las   opciones
correspondientes*/
  if (argc<3) {
    printf ("Use %s cifrado claro\n",argv[0]);
    exit(-1);
  }
  if ((dfp = fopen(argv[1],"r")) == NULL) {
    perror (argv[1]);
    exit(-1);
  }
  if ((dfp1 = fopen(argv[2],"r")) == NULL) {
    perror (argv[2]);
    exit(-1);
```

```
  }


 // de leerachivo en var
    fseek (dfp,0L,SEEK_END);
    lTamano = ftell (dfp);
    sArchivote = malloc (lTamano);
    fseek (dfp,0L,SEEK_SET);
    fread (sArchivote,1,lTamano,dfp);
    fclose (dfp);
// de leerarchivo en var/

 // de leerachivo en var
    fseek (dfp1,0L,SEEK_END);
    lTamano1 = ftell (dfp1);
    sArchivote1 = malloc (lTamano1);
    fseek (dfp1,0L,SEEK_SET);
    fread (sArchivote1,1,lTamano1,dfp1);
    fclose (dfp1);
// de leerarchivo en var/

   printf ("Inicia\n");

/* Si no localoza la llave se imprime un mensaje*/
  if (!rc4_test(sArchivote,sArchivote1,lTamano,palabra,40))
     printf ("No se localizó la llave!\n");

  fclose(dfp2);

}

int rc4_test (const char *sArchivote, const char
*sArchivote1, int lTamano, char *palabra, int nivel)
{
  char *sArchivote2;
  unsigned char c;
  RC4_KEY key;  //se define la varialbe key
```

```
  if (nivel>1)
  {
    for  (c=32;c<126;c++)    {  /*  Se  hace  la  búsqueda
exhaustiva  */
        memset (&palabra[40-nivel],0,nivel);
        palabra[40-nivel]=c;
        if
(rc4_test(sArchivote,sArchivote1,lTamano,palabra,nivel-1))
           return (-1);
    }
  } else  {
    sArchivote2 = malloc (lTamano);
    for (c=32;c<126;c++)
    {
     palabra[39]=c;
     palabra[40]=0;

     RC4_set_key(&key,40,palabra);
     RC4(&key,  lTamano ,sArchivote,sArchivote2);

     if (memcmp(sArchivote2,sArchivote1,lTamano)==0)
      {
/* Imprime la palabra calve así como el texto descifrado*/
        printf("La clave es: %s \n", palabra);
        puts (sArchivote2);
        return -1;
      }
    }
    free(sArchivote2);
  }
  return(0);
}
```

5.4.4 Ataque paralelizado

Paralelizar un programa nos ayuda a ejecutarlo en varios procesadores a la vez, este método es utilizado para correr programas ya sea en una supercomputadora o en un cluster, en este caso el programa fue ejecutado en dos clusters que son malicia y mixbaal cuyas características se pondrán mas adelante.

En éste trabajo y por cuestiones de investigación se concreto sólo a paralelizar el programa `por_fuerza_bruta_a_40.c` ya que no tiene sentido paralelizar el programa que cifra los archivos ni el programa que hace un ataque por diccionario ya que estos programas no ameritan una paralelización pues no necesitan mucho tiempo de ejecución a menos que el archivo de texto sea demasiado grande como sería por ejemplo del orden de terabytes cuestión que este trabajo no cubre y solo nos concretaremos al cifrado de un archivo pequeño por cuestiones prácticas.

La paralelización de un programa consta principalmente de fragmentar en pequeñas partes las tareas que debe realizar el algoritmo del programa asignando las tareas fragmentadas a cada uno de los procesadores con los que se cuenta, por eso es importante saber el número de procesadores con los que se contará para la ejecución del programa en un cluster para así poder fragmentar dichas tareas entre el numero de procesadores a ejecutarse. Esto nos ahorra tiempo de proceso ya que en teoría si un procesador toma un tiempo fijo X en terminar una tarea, M procesadores iguales tomaran X/M en tiempo para realizar la misma tarea por lo que entre mas procesadores se tengan mas rápido se podrá encontrar la solución de un problema

La compilación de un programa paralelizado se realiza de una manera un poco distinta a la del programa a un procesador y es diferente al programa ejecutado con un solo procesador por lo que se indica a continuación la manera en la que se realizó el programa así como se debe ejecutar en un cluster linux, (Las características del cluster se mencionan en la sección 5.2.3) ya que para poder correr el programa por cuestiones de logística del departamento de Super Cómputo de DGSCA hay que encolarlo en alguna de las colas que tiene dicho cluster.

Para compilarlo se ejecuta la siguiente linea:

/local/mpich/bin/mpicc ataque_mpi.c -o ataque_mpi –lssl
La cual utiliza el comando mpicc para realizar la compilación para varios procesadores

del código fuente ataque_mpi.c y lo ponga en el ejecutable ataque_mpi mediante el uso de las bibliotecas de openssl

Para poder encolar el proceso se procede de la siguiente manera:

qsub ataque_mpi.pbs

Para saber si se encoló correctamente:

qstat -a

El archivo ataque_mpi.pbs es un script de shell de unix que se describe a continuación:

5.4.4.1 Script ataque_mpi.pbs

Una vez que el programa se encuentra paralelizado y compilado este se manda a encolar por medio de un script en este se le dice el nombre del programa a encolar, que en nuestro caso es ataque_mpi, se le da el archivo de a donde va a direccional la salida, los errores así como la cantidad de nodos en los que correrá el programa. Además como es importante medir el tiempo se necesita saber a que hora y fecha comenzó el programa y a que hora y fecha se termino

```
#! /bin/sh
#PBS -q prueba
#PBS -e salida.err
#PBS -o salida.out
#PBS -l nodes=6:ppn=2
#PBS -N ataque_mpi

cd $PBS_O_WORKDIR
export LD_LIBRARY_PATH=/usr/lib
echo Corriendo en host `hostname`
echo Fecha Inicio `date`
echo Directorio `pwd`
/usr/bin/time mpiexec -n 12 ./ataque_mpi cifrado claro
echo Fecha Termino `date`
echo Directorio `pwd`
```

La línea que dice: #PBS -l nodes=6:ppn=2 significa 6 nodos y 2 procesadores por nodo. Por logística no se pueden pedir más de 12 procesadores.

Las colas disponibles se ven con:
qstat -q

Lo que da una salida parecida a esto:

```
Queue            Memory CPU Time Walltime Node Run Que Lm  State
---------------- ------ -------- -------- ---- --- --- --  -----
paralela          6000mb 480:00:0    --       6   0   0 --   E R
serial            750mb 122:00:0    --       1   1   0 --   E R
pp               6000mb 480:00:0    --       4   0   0 --   E R
prueba             --   900:00:0    --      16   4   0 --   E R
grids              --     --        --      20   0   0 --   E R
                                           --- ---
                                            5   0
```

5.4.4.2 Programa ataque_mpi.c

A continuación se despliega el listado del programa en paralelo para realizar la búsqueda de la llave por medio de fuerza bruta a 40 bits, que básicamente hace lo mismo que el `fuerza_bruta_40.c` pero de manera paralelizada.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <openssl/rc4.h>

#include <mpi.h> /*Además se incluyen las librerías de
mpi*/


//ataque_mpi.c 27 Noviembre 2005

//Se hace un ataque por fuerza bruta a 40 bits paralelizado
```

```
//Se da el archivo cifrado a revisar así como elarchivo a
comparar


int rc4_test (const char *sArchivote, const char
*sArchivote1, int lTamano, char
*palabra, int nivel, int inicio, int fin);


// MPI

int myid, numprocs;


int main(int argc, char *argv[])

{

//  MPI Se definen las variables necesarias

int inicio, fin, proc,res;

MPI_Request peticion;


//

char palabra[80], nombre[25];

char *c;

//


//Del archivo a cifrar

 char nombre_archivo[50];

 char *sArchivote;

 char *sArchivote2;

 long lTamano;

//De archivo a cifrar
```

```
// Del archivo en texto plano a comparar con el cifrado
  char nombre_archivo1[50];
  char *sArchivote1;
  long lTamano1;


  FILE    *dfp;         //Archivo a analizar
  FILE    *dfp1;      //Archivo a compara en texto plano



  printf ("Calling MPI_Init\n");


  MPI_Init(&argc,&argv);  // Inicia el ambiente de MPI
  printf ("After MPI_Init\n");
  MPI_Comm_size(MPI_COMM_WORLD,&numprocs);  // determina el
num de procesadores
  MPI_Comm_rank(MPI_COMM_WORLD,&myid); // determina el # de
proceso


  printf ("parsing arguments\n");


  if (argc<3) {
    printf ("Use %s cifrado claro\n",argv[0]);
    MPI_Finalize();   // Termina el proceso
    exit(-1);
  }
```

```
  // Todos los procesos leen los archivos por lo que deben
ser accesibles
  // a todos los nodos.


  if ((dfp = fopen(argv[1],"r")) == NULL) {
    perror (argv[1]);
    MPI_Finalize();   // Termina el proceso
    exit(-1);
  }
  if ((dfp1 = fopen(argv[2],"r")) == NULL) {
    perror (argv[2]);
    MPI_Finalize();   // Termina el proceso
    exit(-1);
  }



 // de leerachivo en var
    fseek (dfp,0L,SEEK_END);
    lTamano = ftell (dfp);
    sArchivote = (char  *)malloc (lTamano);
    fseek (dfp,0L,SEEK_SET);
    fread (sArchivote,1,lTamano,dfp);
    fclose (dfp);
// de leerarchivo en var/


 // de leerachivo en var
```

```
    fseek (dfp1,0L,SEEK_END);

    lTamano1 = ftell (dfp1);

    sArchivote1 = (char *) malloc (lTamano1);

    fseek (dfp1,0L,SEEK_SET);

    fread (sArchivote1,1,lTamano1,dfp1);

    fclose (dfp1);

// de leerarchivo en var/



/* Se determina la separación de búsqueda de llaves acuerdo
al numero de procesadores */


   inicio = (256/numprocs)*myid;

   fin    = (256/numprocs)*(myid+1)-1;


   if (numprocs-1 == myid ) // Si es el último le toca
terminar

      fin = 255;



   printf ("(%d) Inicia rango %X->%X\n", myid, inicio, fin
);



   if
(!(res=rc4_test(sArchivote,sArchivote1,lTamano,palabra,5,in
icio,fin)))

      printf ("(%d) No se localizó la llave!\n",myid);

   else if (res==1) {

     for (proc = 0 ; proc<numprocs ; proc++) // Avisa al
resto que ha
terminado

       if (myid != proc)
```

```
          MPI_Isend (&myid, 1, MPI_INT, proc, 0,
MPI_COMM_WORLD,
&peticion);

   }

   printf ("(%d) Termina proceso\n",myid);

   MPI_Finalize();   // Termina el proceso

}
```

```
int rc4_test (const char *sArchivote, const char
*sArchivote1, int lTamano, char
*palabra, int nivel, int inicio, int fin)

{

  // MPI

  int bandera,proc,res;

  MPI_Status estado;

  //

  char *sArchivote2;

  int c;

  RC4_KEY key;  //se define la varialbe key


/*Por cada nodo se realiza la búsqueda correspondiente*/

  if (nivel>1)

  {

    for (c=inicio;c<=fin;c++)  {

        memset (&palabra[5-nivel],0,nivel);

        palabra[5-nivel]=c;
```

```
        if
(res=rc4_test(sArchivote,sArchivote1,lTamano,palabra,nivel-
1,0,255))


           return (res);

    }

  } else  {

    sArchivote2 = (char *)malloc (lTamano);

    for (c=inicio;c<=fin;c++)

    {

     palabra[4]=c;

     palabra[5]=0;



     RC4_set_key(&key,5,(const unsigned char*)palabra);

     RC4(&key,  lTamano ,(const unsigned
char*)sArchivote,(unsigned
char*)sArchivote2);



     if (memcmp(sArchivote2,sArchivote1,lTamano)==0)

      {

        puts (sArchivote2);

        printf("(%d) La clave es: %s \n", myid, palabra);

        return 1;

      }



      for (proc=0; proc<numprocs; proc++)

         if (proc!=myid) {
```

```
            MPI_Iprobe (proc, 0, MPI_COMM_WORLD, &bandera,
&estado);

            if (bandera) {

               printf ("(%d) El proceso %d encontro la
llave!\n",myid,proc);

               return -1;

            }

         }



    }

    free(sArchivote2);

  }


  if (nivel==5) printf ("(%d) last key =
%s\n",myid,palabra);

  return(0);

}
```

5.5 Características de los clusters

Estos programas fueron corridos sobre varias plataformas para conocer el desempeño de las mismas en función del tiempo a continuación se muestran las diferentes plataformas para finalmente llegar a los resultados y conclusiones finales.

Las condiciones en general de los sistemas son:

- Un ambiente tipo UNIX o algún clon de este sistema.
- Compilado por compilador tipo ANSI C (gcc)
- Librerías de openssl (ssl/rc4.h)
- Algunas características más propias de cada cluster que se pondrán en cada caso.

5.5.1 Características del cluster malicia:

1 Nodo maestro: Pentium 4 HT @ 3.2GHz
16 KB cache
512 MB en RAM

6 Nodos esclavos: Pentium 4 @ 2.4 GHz
512 KB
512 MB en RAM

Switch 3Com FastEthernet 100 MHz 24 puertos

Almacenamiento paralelo PVFS con 147 GB

Compiladores Intel y GNU

Biblioteca para envío de mensajes MPI y numéricas LAPACK

5.5.2 Características de cluster mixbaal

Tipo Beowulf

Nodo maestro
2 Procesadores Pentium III a 1130.497 MHz
512KB de cache
1GB de memoria RAM
1 disco duro SCSI de 18 GB
Mother board SCB2
2 tarjetas de red Intel PRO/100

Nodos esclavos
18 mother boards duales SCB2, cada una con:
Dos procesadores Pentium III a 1130.497MHz
512KB de cache
1GB de memoria RAM
Disco duro IDE de 40Gb
2 tarjetas de red Intel PRO/100

Switch Extreme networks Summit 24
Fast Ethernet 100 MHz

24 puertos

Sistema Operativo

Red Hat 7.2

Compiladores y Lenguajes de Programacion GNU compiler 3.1

- gcc
- g77
- g++

The Portland Group 3.3-2

- pgcc
- pgf77
- pgf90
- pgCC
- pgprof

Intel C++ Compiler 6.0

- iccbin
- ifcbin

Absoft

- f77
- f90

Perl 5.6.0 Depuradores
Xpgdbg
ddd 3.3

Bibliotecas Numericas Atlas Bibliotecas Paralelas
mpich-1.2.4
pvm3

Herramientas de Uso General

GNU Make 3.79.1
Plot graphs Gnuplot 7.0
Emacs 20.7-41
Tar 1.13.19-6
Zip 2.3-10
GNU Ghostscript 6.51
Sed 3.02-10

Herramientas de Seguridad

Openssl 0.9.6b-8
Openssh 2.9p2-7
Tcp_wrappers 7.6-19

Shells
Sh
Ksh
Csh
Tcsh
Bash
Sistemas de Colas
  PBS
Sistemas de Archivos
  Pvfs 1.5.3
Network File System 0.3.1-13

NOTA: Las características de las máquinas fueron tomadas en el caso del cluster mixbaal de la pagina http://www.super.unam.mx y las de malicia a petición expresa mía por lo que mantuve la información integra de la forma de configuración y hardware contenido en cada uno de los clusters.

Resultados

Tiempos en malicia

En un procesador Pentium IV @ 2 GHz la prueba de una llave toma aproximadamente: 1.94 E-05 segundos por lo que teóricamente, evaluar todas las llaves posibles que son (256^5) de 40 bits tomaría: (240^5) * 1.94 E-5 seg = 21.256 E+06 seg es decir con un CPU se tardaría:

5,904.68 hrs ~ 246 días aproximadamente.
Por lo que con 5 procesadores (malicia): 49 días aproximadamente.

Los tiempos son máximos por lo que pueden se menores y cuando la llave se encuentre más cerca del inicio en el barrido de las llaves hay más posibilidades de que la solución se encuentre más rápido. Pero eso es cuestión de la manera en la que se barra el espacio de llaves y del tipo de llave empleada para el cifrado.

Datos de mixbaal:

256^4*10 claves en 1 proc: 1118124.24 s ~ 311 hrs ~ 13 dias
tiempo x clave : 1118124.24/(256^4*10)= 26.03E-6 s = 26.03 us
tiempo x 256^5 claves : 26.03E-6 * 256^5 = 28,623,980 s
~ 477,066 min
~ 7,951 hr
~ 331 dias
~ 11 meses

Por lo tanto solo se divide las cantidades anteriores entre el número de procesadores (16 en este caso) y se obtiene el tiempo real.

28,623,980 s / 16 = 1,788,998 s
~ 29,816 min
~ 496 hr
~ 20 dias

La probabilidad de hallar la clave con un procesador es de 1/256^5, y con 16 procesadores es 1/(256/16)^4 = 1/16^4 por lo que la probabilidad de encontrar la clave aumenta considerablemente a medida que se incrementa el número de procesadores, esta es una buena razón para utilizar una versión paralela.

Con una clave a 8 bits la cual fue la letra “a” el programa tardó 74 horas en 16 procesadores dando un total de:

74 hrs * 16 cpus = 1184 hrs de CPU

Con una clave a 40 bits que en realidad fue truncada los resultados fueron los siguientes: Las clave real fue ferrocarrilero pero como se utilizan solamente los primeros 40 bits esta clave se trunca a los primeros 5 caracteres que tiene la clave, que es en este caso ferro, esta clave fue la que encontró el programa.

La corrida inicio en Tue Jan 31 16:59:41 CST 2006 y termino en Fri Feb 3 19:13:17 CST 2006

Clave	a		
T. CPU	57 horas	~	2.7 días
T. CPU total	1184 horas	~	49 días
T. Pared	74:13 horas	~	3 días

Este programa comenzó el Wed Feb 8 17:03:06 CST 2006 y termino en Tue Feb 28 08:58:45 CST 2006 por lo que tardo aproximadamente unos 20 días (un poco menos) en encontrar la clave de cifrado.

Clave:	ferro		
T. CPU:	471 hrs	~	19 días
T. CPU total:	7543 hrs	~	314 días
T. pared:	455:71 horas	~	20 días

El tiempo de pared es el tiempo real en que se tuvo que esperar para obtener
el resultado, el tiempo de CPU es el tiempo que realmente se proceso el programa, el resto del tiempo lo ocupó el procesador en ejecutar procesos del S.O. que es aproximadamente el 30% del total del tiempo de pared.

El ataque por diccionarios se realizó de al siguiente manera
El texto se cifro con el programa cifra_40 y se ataco con el programa por_diccionario, el diccionario se formo con varios diccionarios bajados de la red [ARG06] teniendo un total de 4594387 obtenidas mediante el comando wc –l de UNIX,

```
mlgz@mixbaal:~/tesis/final/finalreal$ wc -l  /tmpu/mlgz/conjuntados/todos
4594387 /tmpu/mlgz/conjuntados/todos
```

Los diccionarios conjuntados en este gran diccionario fueron con los siguientes criterios de palabras:

Croatian	kjbible	n_family	names_hp	swahili	
abbr	danish	Koran	n_fast	norse	swedish
acr-diag	descript.ion	latin	n_femal2	norwegia	tech
aeneid	dogs	lcarrol	n_female	numbers	tolkien.wor
afr_dbf	drugs	microalg	n_finnis	odyssey	trek
algae	dutch	minix	n_french	oz	ul_words
all-word	english	movie-ch	n_given	paradise.los	us-count
allwords	ethnolog	movies	n_given2	phrases	usenet-l
arthur	fable	movies2	n_hindu	places	usenet-m

asteroid	famous	mts	n_male	polish	usenet-n
bacteria	finnish	music-cl	n_male2	python	uunet-si
biology	french	music-co	n_mgerma	rock-gro	viruses
cartoon	fungi	music-ja	n_names	russ_koi	web2
cartoons	german	music-ot	n_norweg	russian_	web2a
charlema	germanl	music-ro	n_other	sf	world_fa
chars	hosts	music-sh	n_people	sg_words	yiddish
chinese	hosts-tx	myths-le	n_people.1	shakesp-	zipcodes
colleges	iliad	n_actor2	n_stati2	shakespe	
common-p	inet-mac	n_actors	n_statis	sindarin	
congress	italian	n_anglos	n_surna2	spanish	
cracklib	japanese	n_chines	n_surnam	sports	
crl-name	junk	n_common	n_swed	statisti	

El criterio para encontrar el tiempo máximo en el diccionario fue el siguiente:

Teniendo en cuenta que el numero de contraseñas es muy reducido se escogió la clave 2828 para cifrar el texto claro, se busco la clave en el programa mediante grep 2828 conjuntado no encontrando ninguna clave como esta, se procedió a agregarla al final del archivo, y como la lectura de este diccionario es lineal de arriba abajo, la última clave sería la que se busca.

Con este criterio se corre el programa:

El tiempo empleado en encontrar la llave por diccionario por medio del comando time fue el siguiente:

mlgz@mixbaal:~/tesis/final/finalreal$ time ./diccionariof cifr texto /tmpu/mlgz
/conjuntados/todos > corrida

real 0m45.636s
user 0m41.830s
sys 0m1.420s

Esta corrida se realizo en la maquina mixbaal y las corridas varían según el hardware que tenga cada maquina.

A continuación se muestra en la tabla R.1 el resumen de los tiempos aproximados reales según las corridas con un procesador, el tiempo que tardaron con los 16 procesadores con los que se realizaron las corridas, así como una proyección a 32, 64 y 128 procesadores para poder tener una visión del tiempo que tardaría un romperse una clave de estas dimensiones de manera real.

Procesadores	1	16	32	64	128
Clave con a	49.33 días	4 días	2 días	1 día	1/2 día
Clave con ferro	324 días	29 días	10 días	5 días	2.5 días

Tabla R.1

NOTA: Las corridas completas con “a” y con “ferro” en mixbaal se encuentran en los apéndices 1 y 2 respectivamente

Discusión y Conclusiones

El algoritmo RC4 es un buen algoritmo de cifrado pero desgraciadamente los 40 bits utilizados para cifrar no son suficientes para mantener el secreto por más de 20 días naturales. Dependiendo del tipo de cluster y de la carga de trabajo que realice en ese momento.

Las claves débiles en información cifrada pueden ser encontradas muy rápidamente por medio de un ataque por diccionario, por lo que un diccionario de 4 millones de palabras tomaría en promedio alrededor de un minuto, probando cada una de las palabras dependiendo de la máquina en la que se realiza la corrida. En este caso no importa el tamaño de la palabra si no la facilidad para encontrarla por medio de un diccionario.

Aunque se tuviera un algoritmo muy fuerte de cifrado, este de nada serviría pues las claves son débiles.

En resumen se debe tener un buen algoritmo de cifrado además de tener claves fuertes de cifrado. El algoritmo RC4 es un buen algoritmo de cifrado ya que ha demostrado ser en la práctica bastante robusto en todos los años que lleva activo como algoritmo comercial. Por ejemplo el algoritmo de cifrado DES ha sido roto y se puede romper cualquier tipo de clave sin importar los bits a que se cifre ni el tamaño que tenga la clave. Cosa que no ocurre con RC4 a la fecha.

Este algoritmo a 40 bits es muy débil y debe manejarse con precaución los datos cifrados a esta cantidad de bits ya que se tiene una falsa confianza de seguridad. El estándar internacional para la exportación de cifrado para RC4 es de 40 bits [ITA96] por lo que se debe replantear el estándar internacional a no ser que por razones de seguridad nacional (dependiendo de cada nación) se mantenga así el cifrado. Pero se debe mantener informados a los usuarios del riesgo que se corre al utilizar el algoritmo con tales características.

La relación de costo y rompimiento de llaves a 40 bits del algoritmo RC4 aún es muy grande, ya que se requiere todavía de varios procesadores para poder romper la llave de cifrado (en este caso 16) y de conocimientos mas profundos para poder armar un cluster estable, o tener una súper computadora con características similares a las de los clusters utilizados, para poder realizar una búsqueda exhaustiva del conjunto completo de llaves en un tiempo razonablemente pequeño.

Referencias

[ACA05] Academia Latino Americana de Seguridad en Cómputo, Unidades 1,2 y 3, 2005, http://www.mslatam.com Revisada el 11 de Junio 2005.

[ARG06] The A.R.G.O.N. All rights reserved. 1998-2006 Fight for your rights and defend your privacy!, http://www.theargon.com/achilles/ Revisada el 4 de Abril del 2006.

[BAK00] Baker Mel, Cryptography Decrypted, Addison Wesley, 2000

[BOR01] Borghello Cristian, Tesis "Seguridad Informática: Sus implicaciones e implementación", Copyright Cristian F Borghello, 2001.

[BOT04] Hugo Bottelier, ¿Robar una contraseña? ¿Que podría ser mas fácil?, http://www.crime-research.org/ Revisada el 17 de Agosto de 2004

[COU04] Couture Nathaniel, Kent Kenneth B, The Effectiveness of brute force attacks on RC4, 2004

[COX05] Cox Mark J, Openssl, 2005, http://www.openssl.org Revisada el 15 de Mayo 2005

[ETB05] ETB, 2005 http://www.etb.net.co Revisada el 11 de Junio de 2005

[FLU04] Fluhrer Scout, Martin Itsik, Shamir Adi, Attack in RC4 an WEP, 2004.

[GAT05] B. Gates, Microsoft Office, 2005 http://officemicrosoft.com Revisada el 11 de Junio de 2005

[GOL98] Golic Jovan Dj, Linear statistical weakness of alleged RC4, Scool of Electrical Engineering, University of Belgrade, 1998.

[GOR05] Gordon Laurece A., Loeb Martin P.,Computer Crime and Security Survey, CSI, 2005.

[GRO00] Grosul Alexander, Wallach Dan, A Related-Key Cryptanalysis of RC4, Department of Computer Science, Rice University.

[ISE04] ISECOM (Institute for Security and Open Methodologies), Lección 11 Passwords, Hacker High School, 2004, http://www.hackerhighchool.org Revisada el 12 de Mayo de 2005

[ITA96] ITAR (International Traffic in Arms Regulations Restriction), Part III, 7.1.1 Dimensions of Choice for Controlling the Export of Cryptography, http://www.jya.com/nrc07.txt Revisada el 6 de Abril de 2006.

[MEN96] Meneses A, Hand Book of Applied Cryptography, CRC Press, 1996

[NIS04] National Institute of Standards and Technology, http://www.nist.gov , Revisada el 21 de Marzo del 2000

[POM03] Pomerans Hal, Common Unix Vulnerabilities, Deer Run Associates, 2003 Track 6 Securing UNIX SANS Institute, http://www.sans.org, Revisada el 30 de Abril de 2004

[RAM04] Ramió Aguirre Jorge, Libro Electrónico de Seguridad Informática y Criptografía, Universidad Politécnica de Madrid España Versión v3.2, 2004

[RIV04] Rivest R, What is a stream cipher? http://www.rsasecurity.com, Revisada el 22 de Mayo de 2004

[SAN03] SANS Institute, Track 1 Essentials with CISSP (Certified Information Systems Security Professionals), 2003 http://www.sans.org, Revisada el 30 de Abril de 2004

[SCI00] Science Daily, http://sciencedaily.com, Revisada el 22 de Febrero del 2000

[SIL00] Silvani Alicia, Recomendaciones para tener contraseñas fuertes, cybsec, 2005, http://www.cybsec.com , Revisada el 11 de Junio 2005

[STE94] Sterndark David (Anónimo), Publicación del algoritmo RC4 (En código fuente), Grupo de noticias: Sci crypt, 1994

[WIK05] Wikipedia, RC4 (cipher), 2005, http://en.wikipedia.org, Revisada el 22 de Enero 2005

GLOSARIO

A

Acceso legal: Acceso por parte de terceras personas o entidades, incluyendo gobiernos, al texto en claro, o claves criptográficas, o datos cifrados, de acuerdo a ley.

Algoritmo: Un proceso definido sin ambigüedades o un conjunto de reglas para solucionar un problema en un número finito de pasos. Los algoritmos para encriptar se llaman normalmente *algoritmos de cifrado*.

Algoritmo Asimétrico: Algoritmo que requiere dos claves diferentes, una para cifrar y otra para descifrar. Una se llama clave pública y la otra clave privada. Un ejemplo es el algoritmo RSA.

Algoritmo de Cifrado por Bloques: Son aquellos algoritmos que cifran los mensajes en unidades de tamaño fijo llamados bloques. El tamaño usual de un bloque es de 64 bits. Ejemplos de estos algoritmos son DES e IDEA.

Algoritmo Simétrico: Algoritmo de cifrado que se caracteriza por usar la misma clave para cifrar y descifrar. Ejemplos de este tipo de algoritmos son DES e IDEA

Algoritmo Hash: Son algoritmos que permiten verificar que un mensaje no ha sido modificado (integridad). Dado un mensaje de tamaño arbitrario, producen una salida de tamaño fijo. Ejemplos de este tipo de algoritmo son MD5 y SHA.

Autenticación: Proceso por el cual se garantiza que el usuario que accede a un sistema de ordenador es quién dice ser. Por lo general, los sistemas de autenticación están basados en el cifrado mediante una clave o contraseña privada y secreta que sólo conoce el auténtico emisor.

Autoridad certificadora: Entidad que da testimonio de la pertenencia o atribución de una determinada firma digital a un usuario o a otro certificador de nivel jerárquico inferior.

B

Bloque: Unidad básica de información que puede ser cifrada o descifrada. En algoritmos simétricos, el tamaño usual de un bloque es de 64 bits.

Browser: Navegador, visualizador; programa o aplicación para navegar a través del Web (WWW), tal como Netscape o Internet Explorer, accediendo a documentos, imágenes, ficheros, etc.

C

Certificado: Documento digital que identifica a la autoridad certificadora que lo ha

emitido, identifica al firmante del mensaje o transacción, contiene la clave pública del firmante, y contiene a su vez la firma digital de la autoridad certificadora que lo ha emitido.

Cifrado: Transformación de un mensaje en otro, utilizando una clave para impedir que el mensaje transformado pueda ser interpretado por aquellos que no conocen la clave.

Clave criptográfica (Llave criptográfica): Parámetro que se utiliza junto con un algoritmo criptográfico para transformar, validar, autenticar, cifrar o descifrar datos.

Clave Publica (Llave Pública): La clave disponible públicamente en un sistema criptográfico de Clave Pública, usado para cifrar mensajes destinados a su propietario y para descifrar firmas hechas por su propietario.

Clave Privada (Llave Privada): La clave secreta de un sistema criptográfico de Clave Pública, usada para descifrar los mensajes entrantes y firmar los salientes.

Confidencialidad: Característica o atributo de la información por el que la misma sólo puede ser revelada a los usuarios autorizados en tiempo y forma determinados.

Control de Acceso: La restricción en el acceso al entorno de una red. En el contexto de Apache significa normalmente la restricción en el acceso a ciertas URLs.

Criptoanálisis: Rama de la ciencia que estudia las técnicas por las cuales se pueden neutralizar o quebrar algoritmos criptográficos. En particular se aplica a la obtención de textos planos a partir de la intercepción de textos cifrados.

Criptología: Rama de la ciencia que comprende a la Criptografía y al Criptoanálisis.

Criptografía: Ciencia que mediante el tratamiento de la información, protege a la misma de modificaciones y utilización no autorizada. Utiliza algoritmos matemáticos complejos para la transformación de la información en un extremo y la realización del proceso inverso en el otro extremo.

Criptográfia de Clave (Llave) Pública: El estudio y aplicación de sistemas de cifrado asimétricos, que usa una clave para cifrar y otra para descifrar. Una clave de cada uno de estos tipos constituye un par de claves. También se llama Criptografia Asimétrica.

D

Datos: Representación de hechos, conceptos o instrucciones bajo una forma adaptada a la comunicación, a la interpretación o al tratamiento por seres humanos o máquinas.

Datos personales: Cualquier información referente a una persona identificada,

Depositario de la clave: Persona o entidad que está en posesión o tiene el control de las claves criptográficas. El depositario de la clave no es necesariamente el usuario de la

misma.

Descifrado: Función inversa al cifrado.

Disponibilidad: El hecho de ser accesibles y utilizables los datos, informaciones o sistemas de información en el tiempo deseado y del modo requerido.

E

Encriptación: Acción de proteger la información mediante técnicas criptográficas ante modificaciones o utilización no autorizada.

F

Firma digital: Información añadida o transformación cifrada de los datos que permite al receptor de los mismos comprobar su fuente e integridad y protegerse así de la suplantación o falsificación. Consiste en una transformación de un mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posea el mensaje inicial y la clave pública del firmante, pueda determinar de forma fiable si dicha transformación se hizo utilizando la clave privada correspondiente a la clave pública del firmante, y si el mensaje ha sido alterado desde el momento en que se hizo la transformación.(Utah) Es un sello integrado en datos digitales, creado con una clave privada, que permite identificar al propietario de la firma y comprobar que los datos no han sido falsificados (Alemania).

H

Hash. Un algoritmo criptográfico que aplica una función de un solo sentido cuya característica es el tomar un valor único por cada conjunto de datos distintos lo que permite diferenciar entre datos que han sido modificados y los que no.

HTML (HyperText Markup Language): Lenguaje en el que se escriben los documentos a los que se acceden mediante los navegadores WWW. Admite componentes hipertexto y multimedia.

HTTP (HiperText Transfer Protocol): Protocolo de Transmisión Hipertexto. Protocolo de comunicaciones utilizado por los programas clientes y servidores de WWW para comunicarse entre sí.

HTTPS: Protocolo de transferencia de Hipertext (Seguro), es el mecanismo de comunicación de cifrado estándar en World Wide Web. En realidad es HTTP sobre SSL.

I

Informaciones: Significado que toman los datos de acuerdo con convenciones vinculadas a estos datos.

Integridad: Garantía de la exactitud de la información frente a la alteración, pérdida o destrucción, ya sea de forma accidental o fraudulenta.

Interoperabilidad: Interoperabilidad de métodos criptográficos es la capacidad técnica de que varios métodos criptográficos funcionen, conjuntamente.

M

Mensaje: Cualquier información, ya sea un archivo o una cadena de caracteres.

Métodos criptográficos: abarca las técnicas, servicios, sistemas, productos y sistemas de gestión de claves criptográficas.

N

Navegador. Véase Browser.

No repudio: Propiedad que se consigue por medios criptográficos, que impide a una persona o entidad negar haber realizado una acción en particular relativa a datos (como los mecanismos de no rechazo de autoría (origen); como demostración de obligación, intención o compromiso; o como demostración de propiedad).

Notario electrónico (TTP, Trusted Third Parties): Entidad pública o privada encargada de la emisión de certificados digitales que atestigüen la autenticidad de los propietarios de los mismos.

O

OpenSSL: El toolkit Open Source para SSL/TLS

Operadores de red: Entidad pública o privada que haga disponible la utilización de una red de telecomunicación.

P

Pass Phrase: La palabra o frase que protege los archivos de clave privada. Evita que usuarios no autorizados los cifren. Normalmente es solo la clave de cifrado/descifrado usada por los Algoritmos de Cifrado.

PGP (Pretty Good Privacy): Programa de libre distribución, escrito por Phil Zimermann, que impide, mediante técnicas de criptografía, que ficheros y mensajes de correo electrónico puedan ser interpretados por personas no autorizadas. Puede también utilizarse para firmar electrónicamente un documento o un mensaje, realizando así la autenticación del autor.

Proveedores de acceso: Organizaciones que suministran la infraestructura técnica necesaria para que los usuarios puedan conectarse a Internet. Para usuarios domésticos, lo habitual es utilizar una conexión a través de la red telefónica básica mediante un módem.

Proveedores de contenido: Personas u organizaciones que publican información de cualquier tipo en Internet, ya sea utilizando recursos propios o los suministrados por un proveedor de acceso.

S

Servidor Web: Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario.

Servidor Web seguro: Servidor Web que utiliza protocolos de seguridad (SSL, SHTTP o PCT) el ejecutar transacciones en él. Un protocolo de seguridad utiliza técnicas de cifrado y autenticación como medios para incrementar la confidencialidad y la fiabilidad de las transacciones.

SET (Secure Electronic Transactions): Protocolo creado para proporcionar mayor seguridad a los pagos on-line con tarjetas de crédito verificando la identidad de los titulares de las tarjetas con "certificados digitales" y cifrado los números de las tarjetas durante todo el trayecto, desde el navegante, el vendedor y el centro de proceso de datos. Este estándar ha sido creado por VISA y Master Card y tiene un amplio apoyo de la comunidad bancaria mundial.

Sistema de gestión de claves: Sistema para la generación, almacenamiento, distribución, revocación, eliminación, archivo, certificación o aplicación de claves criptográficas.

Sistemas de información: Ordenadores, instalaciones de comunicación y redes de ordenadores y de comunicación, así como los datos e informaciones que permiten conservar, tratar, extraer o transmitir, incluidos los programas, especificaciones y procedimientos destinados a su funcionamiento, utilización y mantenimiento.

SSL (Secure Sockets Layer): Protocolo, creado por Netscape, para crear conexiones seguras al servidor, de tal modo que la información viaja cifrada a través de Internet.

T

TCP/IP (Transmision Control Protocol/Internet Protocol): Conjunto de protocolos que definen Internet, permitiendo que diferentes tipos de ordenadores - con diferentes sistemas operativos - se comuniquen entre sí.

Texto cifrado: El resultado de haber aplicado a un texto sin cifrar un algoritmo de cifrado.

Texto en Claro: Idem texto plano

Texto plano: Un texto no cifrado.

V

Vector de Inicialización Conjunto de bits aleatorios utilizados en modos de operación retro-alimentados de forma tal que un mismo mensaje se cifre siempre distinto. Estos vectores no tienen que permanecer secretos y pueden ser transmitidos junto con el mensaje cifrado.

W

World Wide Web (WWW, Web, W3): Sistema de información global distribuido, desarrollado por investigadores del CERN en Suiza, que utiliza el protocolo HTTP para enlazar páginas mediante mecanismos de hipertexto (lenguaje HTML).

APENDICE 1

Corrida de mixbaal con la clave a

```
Running on host nodo22
Time is Tue Jan 31 16:59:41 CST 2006
Directory is /home/staff/eml/Siddhartha
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
After MPI_Init
parsing arguments
After MPI_Init
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
(10) Inicia rango A0->AF
(11) Inicia rango B0->BF
(8) Inicia rango 80->8F
(9) Inicia rango 90->9F
```

(12) Inicia rango C0->CF
(13) Inicia rango D0->DF
(15) Inicia rango F0->FF
(6) Inicia rango 60->6F
(7) Inicia rango 70->7F
(4) Inicia rango 40->4F
(5) Inicia rango 50->5F
(2) Inicia rango 20->2F
(3) Inicia rango 30->3F
(0) Inicia rango 0->F
(1) Inicia rango 10->1F
(14) Inicia rango E0->EF
Un virus preocupa más que el terrorismo.- Una encuesta realizada por la
consultora "Harris Interactive" difundida en el marco del Foro
Económico Mundial de Davos Suiza, reveló que actualmente a las personas les
preocupa más el poder perder la información personal o confidencial de su
computadora, a un ataque terrorista, quedarse sin empleo, un desastre natural o
algún tipo de epidemia. Christopher Rodríguez, director general de la firma que
pidió el estudio, señaló que la encuesta analizó las opiniones de los usuarios
en países desarrollados y emergentes, mostrando que hoy en día, la información
es un tema que causa gran preocupación a los consumidores de todo el planeta.

(6) La clave es: a
(6) Termina proceso
(3) El proceso 6 encontro la llave!
(4) El proceso 6 encontro la llave!
(4) Termina proceso
(2) El proceso 6 encontro la llave!
(2) Termina proceso
(5) El proceso 6 encontro la llave!
(5) Termina proceso
(1) El proceso 6 encontro la llave!
(1) Termina proceso
(10) El proceso 6 encontro la llave!
(10) Termina proceso
(9) El proceso 6 encontro la llave!
(9) Termina proceso
(3) Termina proceso
(11) El proceso 6 encontro la llave!
(11) Termina proceso
(7) El proceso 6 encontro la llave!
(7) Termina proceso
(0) El proceso 6 encontro la llave!
(0) Termina proceso
(8) El proceso 6 encontro la llave!
(8) Termina proceso
(12) El proceso 6 encontro la llave!
(12) Termina proceso
(14) El proceso 6 encontro la llave!
(14) Termina proceso
(15) El proceso 6 encontro la llave!
(15) Termina proceso
(13) El proceso 6 encontro la llave!

```
(13) Termina proceso
Time is Fri Feb 3 19:13:17 CST 2006
Directory is /home/staff/eml/Siddhartha
        Command being timed: "mpiexec -n 16 ./ataque_mpi cifrado claro"
        User time (seconds): 1.36
        System time (seconds): 53.80
        Percent of CPU this job got: 0%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 74:13:35
        Average shared text size (kbytes): 0
        Average unshared data size (kbytes): 0
        Average stack size (kbytes): 0
        Average total size (kbytes): 0
        Maximum resident set size (kbytes): 0
        Average resident set size (kbytes): 0
        Major (requiring I/O) page faults: 522
        Minor (reclaiming a frame) page faults: 151
        Voluntary context switches: 0
        Involuntary context switches: 0
        Swaps: 0
        File system inputs: 0
        File system outputs: 0
        Socket messages sent: 0
        Socket messages received: 0
        Signals delivered: 0
        Page size (bytes): 4096
        Exit status: 0
```

APENDICE 2

Corrida de mixbaal con la clave ferro

```
Running on host nodo09
Time is Wed Feb 8 17:03:06 CST 2006
Directory is /home/staff/eml/Siddhartha
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
Calling MPI_Init
After MPI_Init
parsing arguments
After MPI_Init
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
After MPI_Init
parsing arguments
(3) Inicia rango 30->3F
(0) Inicia rango 0->F
(2) Inicia rango 20->2F
(4) Inicia rango 40->4F
(6) Inicia rango 60->6F
```

(5) Inicia rango 50->5F
(8) Inicia rango 80->8F
(10) Inicia rango A0->AF
(12) Inicia rango C0->CF
(14) Inicia rango E0->EF
(1) Inicia rango 10->1F
(7) Inicia rango 70->7F
(9) Inicia rango 90->9F
(11) Inicia rango B0->BF
(13) Inicia rango D0->DF
(15) Inicia rango F0->FF
Un virus preocupa más que el terrorismo.- Una encuesta realizada por la
consultora "Harris Interactive" difundida en el marco del Foro
Económico Mundial de Davos Suiza, reveló que actualmente a las personas les
preocupa más el poder perder la información personal o confidencial de su
computadora, a un ataque terrorista, quedarse sin empleo, un desastre natural o
algún tipo de epidemia. Christopher Rodríguez, director general de la firma que
pidió el estudio, señaló que la encuesta analizó las opiniones de los usuarios
en países desarrollados y emergentes, mostrando que hoy en día, la información
es un tema que causa gran preocupación a los consumidores de todo el planeta.

(6) La clave es: ferro
(6) Termina proceso
(7) El proceso 6 encontro la llave!
(4) El proceso 6 encontro la llave!
(2) El proceso 6 encontro la llave!
(5) El proceso 6 encontro la llave!
(7) Termina proceso
(12) El proceso 6 encontro la llave!
(3) El proceso 6 encontro la llave!
(4) Termina proceso
(8) El proceso 6 encontro la llave!
(1) El proceso 6 encontro la llave!
(10) El proceso 6 encontro la llave!
(14) El proceso 6 encontro la llave!
(11) El proceso 6 encontro la llave!
(13) El proceso 6 encontro la llave!
(2) Termina proceso
(5) Termina proceso
(12) Termina proceso
(3) Termina proceso
(9) El proceso 6 encontro la llave!
(8) Termina proceso
(10) Termina proceso
(14) Termina proceso
(9) Termina proceso
(11) Termina proceso
(13) Termina proceso
(15) El proceso 6 encontro la llave!
(15) Termina proceso
(0) El proceso 6 encontro la llave!
(0) Termina proceso
(1) Termina proceso

```
Time is Tue Feb 28 08:58:45 CST 2006
Directory is /home/staff/eml/Siddhartha
        Command being timed: "mpiexec -n 16 ./ataque_mpi cifrado claro"
        User time (seconds): 11.47
        System time (seconds): 353.24
        Percent of CPU this job got: 0%
        Elapsed (wall clock) time (h:mm:ss or m:ss): 471:55:38
        Average shared text size (kbytes): 0
        Average unshared data size (kbytes): 0
        Average stack size (kbytes): 0
        Average total size (kbytes): 0
        Maximum resident set size (kbytes): 0
        Average resident set size (kbytes): 0
        Major (requiring I/O) page faults: 522
        Minor (reclaiming a frame) page faults: 153
        Voluntary context switches: 0
        Involuntary context switches: 0
        Swaps: 0
        File system inputs: 0
        File system outputs: 0
        Socket messages sent: 0
        Socket messages received: 0
        Signals delivered: 0
        Page size (bytes): 4096
        Exit status: 0
```