UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ENCRIPTACION Y SEGURIDAD EN INTERNET IMPLEMENTACION DE UNA RED VPN FULL MESH POR INTERNET

INFORME DE SUFICIENCIA

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

JOSE GERMAN CASANA VASQUEZ

PROMOCIÓN
1999-II
LIMA-PERÚ
2005

ENCRIPTACIÓN Y SEGURIDAD EN INTERNET

IMPLEMENTACION DE UNA RED VPN FULL MESH POR INTERNET

Dedico este trabajo a:

Mi padre, inspiración plena de lucha y sacrificio,

Mi Hermana, por el apoyo incondicional en mi carrera,

Y mis sobrinos, esperanza de superación.

—¿Qué significa *habla, amigo y entra*? —preguntó Merry.
—Es bastante claro —dijo Grimli—. Si eres un amigo, dices la contraseña y las puertas se abren y puedes entrar. —Sí —dijo Gandalf—, es probable que estas puertas estén gobernadas por palabras. . .

El Señor de Los Anillos
J.R.R. Tolkien

SUMARIO

En este informe se repasa la importancia en el campo de la criptografía y que han prevalecido hasta hoy. En el capitulo I se incluye todos los conceptos básicos y se introduce la terminología empleada en el resto de los capitulo. Su lectura es recomendable incluso para las personas que ya conocen el tema, puesto que pude evitar confusión en los términos empleados a lo largo de la obra. En el capitulo II se introduce a la Criptografía Clásica es decir de la criptografía de llave privada ó sistemas simétricos así como los algoritmos de cifrado. En el capitulo III se introduce a la Criptografía Moderna es decir de la criptografía de llave publica ó sistemas asimétricos.. En el capitulo IV que es la parte menos teórica y quizás más practica desde el punto de vista de la seguridad (no desde el punto de vista criptográfico). Se estudian en ella brevemente los problemas que se dan en una red para que el lector pueda comenzar a elaborar por si mismo estrategias de protección de la información y el por que de la necesidad de implementar redes VPN, las cuales pueden extenderse a realizarse como una malla-completa (Full-Mesh) y con bajos costos de implementación cuando las conexiones son a través del Internet.

ÍNDICE

PRÓLOGO

Es difícil imaginarse hoy, vivir sin un cajero automático o sin el correo electrónico, por mencionar algunas acciones cotidianas, conectarse o no conectarse no forma parte de ningún cuestionamiento en estos días, todos queriendo o no vivimos conectados.

Hoy en día es imposible pensar que nuestras vidas puedan prescindir de las redes, ni que hablar de las redes de redes, que es la Internet a la cual se conectan miles de personas cada día. Pero poco somos conscientes de cómo debemos cuidar unas de las tecnologías claves de este siglo.

Si nos remontamos a la época por ejemplo la 2da guerra mundial y las demás guerras que han pasado, para atacar a un país, o destruir la economía de un país, se necesitaba armamento muy difícil de conseguir y muy costoso a la vez.

Hoy en día, simplemente con redes de computadoras, que cualquiera las consigue y que esta accesible en cualquier lado y a precios muy bajos se puede destruir una economía. Por esto, cuidar las redes, se ha vuelto en nuestros días un tema de seguridad de estado.

En países del orbe mundial y empresas privadas se han comenzado a crear cargos directivos muy altos de gente dedicada a temas de seguridad.

Los Hackers son los malos de esta película real, en muchas casos lo hacen por su cuenta o por mostrar su malestar. Muchas veces son pagados para buscar la información, es por eso que hoy en día debemos estar concientes de que todo lo que se implemente en infraestructura debe ser completamente asegurado.

CAPÍTULO I

INTRODUCCIÓN

Dos problemas aquejan a los documentos electrónicos: La Confidencialidad y la Autenticidad.

La confidencialidad se refiere a la capacidad de mantener un documento electrónico inaccesible a todos, excepto a una lista determinada de personas.

La autenticidad se refiere a la capacidad de determinar si una lista determinada de personas han establecido su reconocimiento y/o compromiso sobre el contenido del documento electrónico.

Existe una diferencia sutil pero muy importante entre el concepto de autenticidad y el concepto de no-repudiación. Por ejemplo usted puede presenciar que un documento fue escrito por alguien pues lo vio en persona. Si el documento no esta firmado autógrafa mente usted estará absolutamente convencido de su autenticidad pero no podrá probarlo pues sin la firma autógrafa es imposible establecer el vínculo entre la voluntad de la persona y el contenido del documento. Si se puede probar a terceros que efectivamente el documento es autentico entonces se dice que el documento es no-repudiable. Si un documento es no-repudiable es autentico pero no viceversa. Una característica básica de un documento autentico es su integridad. En un documento tradicional como un contrato o cheque, si se aprecian

modificaciones o tachones el documento es prácticamente invalidado. En un documento electrónico en donde por errores de transmisión o fallas en el medio de almacenaje o intencionadamente se modifica el contenido original del documento entonces el documento pierde su integridad y por tanto su autenticidad. Si un documento es autentico entonces es integro pero no viceversa.

Estos problemas, confidencialidad, integridad, autenticidad y no-repudiación se resuelven mediante la tecnología llamada ¨Criptografía¨.

La criptografía es una rama de las matemáticas, que al aplicarse a mensajes digitales, proporcionan las herramientas idóneas para solucionar los problemas antes mencionados. Al problema de la confidencialidad se le relaciona comúnmente con técnicas denominadas de "encripción" y el problema de la autenticidad mediante técnicas denominadas de "firma digital", aunque ambos en realidad se reducen a procedimientos criptográficos de encriptación y desencriptación.

1.1. Algunas notas sobre la historia de la Criptografía.

La Criptografía moderna nace al mismo tiempo que las computadoras. Durante la Segunda Guerra Mundial, en un lugar llamado Bletchley Park, un grupo de científicos entre los que se encontraba Alan Turing, trabajaba en el proyecto ULTRA tratando de descifrar los mensajes enviados por el ejército alemán con los más sofisticados ingenios de codificación ideados hasta entonces: la máquina ENIGMA y el cifrado Lorenz. Este grupo de científicos diseño y utilizó el primer computador de la Historia, denominado Colossus, aunque esta información permaneció en secreto hasta mediados de los 70. La mayor parte de los estudios hasta hace relativamente poco tiempo han sido tratados como secretos militares.

Sin embargo, en los últimos años, investigaciones serias llevadas a cabo en universidades de todo el mundo han logrado que la Criptografía sea una ciencia al alcance de todos, y que se convierta en la piedra angular de asuntos tan importantes como el comercio electrónico, la telefonía móvil o las nuevas plataformas de distribución de contenidos multimedia. Por esto no cabe duda de que la información se está convirtiendo en la mayor fuente de poder que ha conocido la Humanidad, y que la Criptografía es una herramienta esencial para su control. Es necesario, pues, que los ciudadanos de a pie conozcan sus ventajas e inconvenientes, sus peligros y leyendas. Dicen que vivimos en Democracia pero, si a la gente no se le muestra toda la información relevante de manera honesta e imparcial, ¿Cómo va a poder decidir su futuro?.

1.2. Conceptos de Criptografía.

1.2.1. Criptografía.

Según el Diccionario de la Real Academia, la palabra Criptografía proviene del griego "κρυπτός" que significa oculto, y "γράφειν" que significa escritura, y su definición es: "Arte de escribir con clave secreta o de un modo enigmático". Entre las disciplinas que engloba cabe destacar la Teoría de la Información, la Teoría de Números o Matemática Discreta, que estudia las propiedades de los números enteros, y la Complejidad Algorítmica.

Existen dos trabajos fundamentales sobre los que se apoya prácticamente toda la teoría criptográfica actual. Uno de ellos, desarrollado por Claude Shannon en sus artículos "A Mathematical Theory of Communication" (1948) y "Communication Theory of Secrecy Systems" (1949), sienta las bases de la Teoría de la Información y

de la Criptografía moderna. El segundo, publicado por Whitfield Diffie y Martín Hellman en 1976, se titulaba "New directions in Cryptography", e introducía el concepto de Criptografía de Llave Pública, abriendo enormemente el abanico de aplicación de esta disciplina.

1.2.2. Criptosistema.

Definiremos un Criptosistema como una quíntupla (M, C, K, E, D), donde:

a) M: representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto claro, o plaintext) que pueden ser enviados.

b) C: representa el conjunto de todos los posibles mensajes cifrados, o criptogramas.

c) K: representa el conjunto de claves que se pueden emplear en el criptosistema.

d) E: es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave k.

e) D: es el conjunto de transformaciones de descifrado, análogo a E.

Todo criptosistema ha de cumplir la siguiente condición:

$$D_k(E_k(m)) = m \tag{1.1}$$

Lo que implica, que si tenemos un mensaje m, lo ciframos empleando la clave k y luego lo desciframos empleando la misma clave, obtenemos de nuevo el mensaje original m.

Existen dos tipos fundamentales de criptosistemas:

Criptosistemas simétricos ó de clave privada: Son aquellos que emplean la misma clave k tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados en comunicaciones la clave k debe estar tanto en el emisor como en el

receptor, lo cual nos lleva preguntarnos cómo transmitir la clave de forma segura.

Criptosistemas asimétricos o de llave pública: Son aquellos que emplean una doble clave (k_p, K_P). k_p se conoce como clave privada y K_P se conoce como clave pública. Una de ellas sirve para la transformación E de cifrado y la otra para la transformación D de descifrado. En muchos casos son intercambiables, esto es, si empleamos una para cifrar la otra sirve para descifrar y viceversa. Estos criptosistemas deben cumplir además que el conocimiento de la clave pública K_P no permita calcular la clave privada k_p. Ofrecen un abanico superior de posibilidades, pudiendo emplearse para establecer comunicaciones seguras por canales inseguros puesto que únicamente viaja por el canal la clave pública, que sólo sirve para cifrar, o para llevar a cabo autentificaciones.

La seguridad de los criptosistemas se suele medir en términos del número de computadoras y del tiempo necesarios para romperlos, y a veces simplemente en función del dinero necesario para llevar a cabo esta tarea con garantías de éxito.

En cualquier caso hoy por hoy existen sistemas que son muy poco costosos ó incluso gratuitos, como algunas versiones de PGP, y que nos garantizan un nivel de protección tal que toda la potencia de cálculo que actualmente hay en el planeta seria insuficiente para romperlos.

1.2.3. Criptoanálisis.

El criptoanálisis consiste en comprometer la seguridad de un criptosistema. Esto se puede hacer descifrando un mensaje sin conocer la llave, o bien obteniendo a partir de uno o más criptogramas la clave que ha sido empleada en su codificación.

Los métodos que buscan exhaustivamente por el espacio de claves k, se denominan ataques por la fuerza bruta, y en muchos casos no suelen considerarse

como auténticas técnicas de criptoanálisis, reservándose este término para aquellos mecanismos que explotan posibles debilidades intrínsecas en el algoritmo de cifrado. Pero no obstante, existen longitudes de clave para las que resultaría imposible a todas luces un ataque de fuerza bruta. Por ejemplo, si diseñáramos una máquina capaz de recorrer todas las combinaciones que pueden tomar 256 bits, cuyo consumo fuera mínimo en cada cambio de estado, no habría energía suficiente en el Universo para que pudiera completar su trabajo.

Un par de métodos de criptoanálisis que han dado interesantes resultados son el análisis diferencial y el análisis lineal. El primero de ellos, partiendo de pares de mensajes con diferencias mínimas usualmente de un bit, estudia las variaciones que existen entre los mensajes cifrados correspondientes, tratando de identificar patrones comunes.

El segundo emplea operaciones XOR entre algunos bits del texto claro y algunos bits del texto cifrado, obteniendo finalmente un único bit. Si realizamos esto con muchos pares de texto claro-texto cifrado podemos obtener una probabilidad p en ese bit que calculamos. Si p está suficientemente sesgada (no se aproxima a 1/2), tendremos la posibilidad de recuperar la clave. Otro tipo de análisis consistiría en tratar de deducir la llave privada a partir de la pública. Suelen ser técnicas analíticas que básicamente intentan resolver los problemas de elevado coste computacional en los que se apoyan estos criptosistemas: factorización, logaritmos discretos, etc.

1.2.4. Seguridad

El concepto de seguridad en la información es mucho más amplio que la simple protección de los datos a nivel lógico. Para proporcionar una seguridad real

hemos de tener en cuenta múltiples factores, tanto internos como externos. En primer lugar habría que caracterizar el sistema que va a albergar la información para poder identificar las amenazas, y en este sentido podríamos hacer la siguiente subdivisión:

Sistemas aislados: Son los que no están conectados a ningún tipo de red. De unos años a esta parte se han convertido en minoría, debido al auge que ha experimentado Internet.

Sistemas interconectados: Hoy por hoy casi cualquier ordenador pertenece a alguna red, enviando y recogiendo información del exterior casi constantemente. Esto hace que las redes de ordenadores sean cada vez más dinámicas, complejas y supongan un peligro potencial que no puede en ningún caso ser ignorado.

En cuanto a las cuestiones de seguridad que hemos de fijar podríamos clasificarlas de la siguiente forma:

Seguridad física.

Seguridad de la información.

Seguridad del canal de la comunicación.

Problemas de autentificación.

Problemas de suplantación.

No repudio.

CAPÍTULO II

CRIPTOGRAFIA DE LLAVE PRIVADA

2.1. Criptografía Clásica

El ser humano siempre ha tenido secretos de muy diversa índole, y ha buscado mecanismos para mantenerlos fuera del alcance de miradas indiscretas. Julio César empleaba un sencillo algoritmo para evitar que sus comunicaciones militares fueran interceptadas. Leonardo Da Vinci escribía las anotaciones sobre sus trabajos de derecha a izquierda y con la mano zurda. Otros personajes, como Sir Francis Bacon o Edgar Allan Poe eran conocidos por su afición a los códigos criptográficos, que en muchas ocasiones constituían un apasionante divertimento y un reto para el ingenio.

En este capítulo haremos un breve repaso de los mecanismos criptográficos considerados clásicos. Podemos llamar así a todos los sistemas de cifrado anteriores a la II Guerra Mundial, o lo que es lo mismo, al nacimiento de las computadoras. Estas técnicas tienen en común que pueden ser empleadas usando simplemente lápiz y papel, y que pueden ser criptoanalizadas casi de la misma forma. De hecho, con la ayuda de las computadoras, los mensajes cifrados empleando estos códigos son fácilmente descifrables, por lo que cayeron rápidamente en desuso. La transición desde la Criptografía clásica a la moderna se da precisamente durante la II Guerra

Mundial, cuando el Servicio de Inteligencia aliado rompe la máquina de cifrado del ejército alemán, llamada ENIGMA.

Todos los algoritmos criptográficos clásicos son simétricos, ya que hasta mediados de los años setenta no nació la Criptografía asimétrica, y por esa razón este capítulo se engloba dentro del bloque de la asignatura dedicado a los algoritmos de llave privada.

2.1.1. Algoritmos clásicos de cifrado

En esta sección veremos algunos criptosistemas que en la actualidad han perdido su eficacia, debido a que son fácilmente criptoanalizables empleando cualquier computadora doméstica, pero que fueron empleados con éxito hasta principios del siglo XX. Algunos se remontan incluso, como el algoritmo de César, a la Roma Imperial. Sin embargo mantienen un interés teórico, ya que nos van a permitir explotar algunas de sus propiedades para entender mejor los algoritmos modernos.

2.1.1.1 Cifrados Monoalfabéticos

Se engloban dentro de este apartado todos los algoritmos criptográficos que, sin desordenar los símbolos dentro del mensaje, establecen una correspondencia única para todos ellos en todo el texto. Es decir, si al símbolo A le corresponde el símbolo D, esta correspondencia se mantiene a lo largo de todo el mensaje.

Algoritmo de César: el algoritmo de César, llamado así porque es el que empleaba Julio César para enviar mensajes secretos, es uno de los algoritmos criptográficos más simples. Consiste en sumar 3 al número de orden de cada letra. De esta forma a

la A le corresponde la D, a la B la E, y así sucesivamente. Si asignamos a cada letra un número (A = 0, B = 1. . .), y consideramos un alfabeto de 26 letras, la transformación criptográfica sería:

$$C = (M + 3) \text{ mód } 26 \qquad (2.1)$$

Obsérvese que este algoritmo ni siquiera posee clave, puesto que la transformación siempre es la misma. Obviamente, para descifrar basta con restar 3 al número de orden de las letras del criptograma.

Sustitución Afí: es el caso general del algoritmo de César. Su transformación sería:

$$E(a,b)(M) = (aM + b) \text{ mód } N \qquad (2.2)$$

Siendo *a* y *b* dos números enteros menores que el cardinal N del alfabeto, y cumpliendo que mcd(a,N) = 1. La clave de cifrado k viene entonces dada por el par (a, b). El algoritmo de César sería pues una transformación afín con k = (1, 3).

Cifrado Monoalfabético General: es el caso más general de cifrado monoalfabético. La sustitución ahora es arbitraria, siendo la clave k precisamente la tabla de sustitución de un símbolo por otro. En este caso tenemos N! posibles claves.

2.1.1.2. Cifrados Polialfabéticos

En los cifrados polialfabéticos la sustitución aplicada a cada carácter varía en función de la posición que ocupe éste dentro del texto claro. En realidad corresponde a la aplicación cíclica de n cifrados monoalfabéticos.

Cifrado de Vigènere: es un ejemplo típico de cifrado polialfabético que debe su nombre a Blaise de Vigènere, su creador, y que data del siglo XVI. La clave está constituida por una secuencia de símbolos $K = \{k_0, k_1, \ldots, k_{d-1}\}$, y que emplea la siguiente función de cifrado:

$$E_k(mi) = m_i + k(_{i \text{ mód } d}) \text{ (mód n)} \qquad (2.3)$$

Siendo mi el i-ésimo símbolo del texto claro y n el cardinal del alfabeto de entrada.

Para criptoanalizar este tipo de claves basta con efectuar d análisis estadísticos independientes agrupando los símbolos según la k_i empleada para codificarlos. Para estimar d, buscaremos la periodicidad de los patrones comunes que puedan aparecer en el texto cifrado. Obviamente, para el criptoanálisis, necesitaremos al menos d veces más cantidad de texto que con los métodos monoalfabéticos.

2.1.1.3. Cifrados de sustitución homofónica

Para paliar la sensibilidad frente a ataques basados en el estudio de las frecuencias de aparición de los símbolos, existe una familia de algoritmos que trata de ocultar las propiedades estadísticas del texto claro empleando un alfabeto de salida con más símbolos que el alfabeto de entrada. Supongamos que nuestro alfabeto de entrada posee cuatro letras, {a, b, c, d}. Supongamos además que en nuestros textos la letra a aparece con una probabilidad 0.4, y el resto con probabilidad 0.2. Podríamos emplear el siguiente alfabeto de salida {α, β, γ, δ, ε} efectuando la siguiente asociación:

E(a) = { α con probabilidad ½).

E(a) = { β con probabilidad ½).

E (b) = γ; E (c) = δ; E (d) = ε.

En el texto cifrado ahora todos los símbolos aparecen con igual probabilidad, lo que imposibilita un ataque basado en frecuencias. A diferencia de lo que se puede pensar en un principio, este método presenta demasiados inconvenientes para ser útil

en la práctica: además del problema de necesitar un alfabeto de salida mayor que el de entrada, para aplicarlo hace falta conocer la distribución estadística a priori de los símbolos en el texto claro, información de la que, por desgracia, no siempre se dispone.

2.1.1.4. Cifrados de transposición

Este tipo de mecanismos de cifrado no sustituye unos símbolos por otros, sino que cambia su orden dentro del texto. Quizás el más antiguo conocido sea el citado, formado por un bastón cilíndrico con un radio particular y una tira de piel que se enrollaba alrededor de aquél. El texto se escribía a lo largo del bastón y sólo podía ser leído si se disponía de otro bastón de dimensiones similares. Un mecanismo de transposición sencillo, que no precisa otra cosa que lápiz y papel, podría consistir en colocar el texto en una tabla de n columnas, y dar como texto cifrado los símbolos de una columna —ordenados de arriba a abajo— concatenados con los de otra, etc. La clave k se compondría del número n junto con el orden en el que se deben leer las columnas.

Por ejemplo, supongamos que queremos cifrar el texto *"El perro de San Roque no tiene rabo"*, con n = 5 y la permutación {3, 2, 5, 1, 4} como clave. Colocamos el texto en una tabla y obtenemos:

1	2	3	4	5
E	L		P	E
R	R	O		D
E		S	A	N
	R	O	Q	U
E		N	O	
T	I	E	N	E
	R	A	B	O

Tabla 2.1. Concatenación de textos.

Tendríamos como texto cifrado la concatenación de las columnas 3, 2, 5,1 y 4 respectivamente: "Osonealr r irednu eoere et p aqonb". Nótese que en la Tabla 2.1 se

muestra que hemos de conservar el espacio al principio del texto cifrado para que el mecanismo surta efecto.

Este tipo de mecanismos de cifrado se puede criptoanalizar efectuando un estudio estadístico sobre la frecuencia de aparición de pares y tripletas de símbolos en el lenguaje en que esté escrito el texto claro. Suponiendo que conocemos n, que en nuestro caso es igual a 5, tenemos 5! = 120 posibles claves. Descifraríamos el texto empleando cada una de ellas y comprobaríamos si los pares y tripletas de símbolos consecutivos que vamos obteniendo se corresponden con los más frecuentes en castellano. De esa forma podremos asignarle una probabilidad automáticamente a cada una de las posibles claves. Si, por el contrario, desconocemos n, basta con ir probando con n = 2, n = 3 y así sucesivamente. Este método es bastante complejo de llevar a cabo manualmente, a no ser que se empleen ciertos trucos, pero una computadora puede completarlo en un tiempo más que razonable sin demasiados problemas.

2.2. Cifrados por bloques

2.2.1. Cifrado de producto

La gran mayoría de los algoritmos de cifrado simétricos se apoyan en los conceptos de confusión y difusión inicialmente propuestos por Shannon, que se combinan para dar lugar a los denominados cifrados de producto. Estas técnicas consisten básicamente en *trocear* el mensaje en bloques de tamaño fijo, y aplicar la función de cifrado a cada uno de ellos.

Recordemos que la confusión consiste en tratar de ocultar la relación que existe entre el texto claro, el texto cifrado y la clave. Un buen mecanismo de

confusión hará demasiado complicado extraer relaciones estadísticas entre las tres cosas. Por su parte la difusión trata de repartir la influencia de cada bit del mensaje original lo más posible entre el mensaje cifrado.

Hemos de hacer notar que la confusión por sí sola sería suficiente, ya que si establecemos una tabla de sustitución completamente diferente para cada clave con todos los textos claros posibles tendremos un sistema extremadamente seguro. Sin embargo, dichas tablas ocuparían cantidades astronómicas de memoria, por lo que en la práctica serían inviables.

Por ejemplo, un algoritmo que codificara bloques de 128 bits empleando una clave de 80 bits necesitaría una tabla de aproximadamente 1063 entradas.

Lo que en realidad se hace para conseguir algoritmos fuertes sin necesidad de almacenar tablas enormes es intercalar la confusión (sustituciones simples, con tablas pequeñas) y la difusión (permutaciones). Esta combinación se conoce como cifrado de producto. La mayoría de los algoritmos se basan en diferentes capas de sustituciones y permutaciones, estructura que denominaremos Red de Sustitución-Permutación. En muchos casos el criptosistema no es más que una operación combinada de sustituciones y permutaciones, repetida n veces, como ocurre con DES.

2.2.1.1. Cifrados con estructura de grupo

Otra de las cuestiones a tener en cuenta en los cifrados de producto es la posibilidad de que posean estructura de grupo. Se dice que un cifrado tiene estructura de grupo si se cumple la siguiente propiedad:

Para todo k_1, k_2,

Existe k_3 tal que $$E_{k2}(E_{k1}(M)) = E_{k3}(M) \quad (2.4)$$

Esto es, si hacemos dos cifrados encadenados con k_1 y k_2, existe una clave k_3 que realiza la transformación equivalente.

Es interesante que un algoritmo criptográfico carezca de este tipo de estructura, ya que si ciframos un mensaje primero con la clave k_1 y el resultado con la clave k_2, es como si hubiéramos empleado una clave de longitud doble, aumentando la seguridad del sistema. Si, por el contrario, la transformación criptográfica presentara estructura de grupo, esto hubiera sido equivalente a cifrar el mensaje una única vez con una tercera clave, con lo que no habríamos ganado nada.

2.2.1.2. S-Cajas

Hemos dicho antes que para poder construir buenos algoritmos de producto, intercalaremos sustituciones sencillas (confusión), con tablas pequeñas, y permutaciones (difusión). Estas tablas pequeñas de sustitución se denominan de forma genérica S-Cajas.

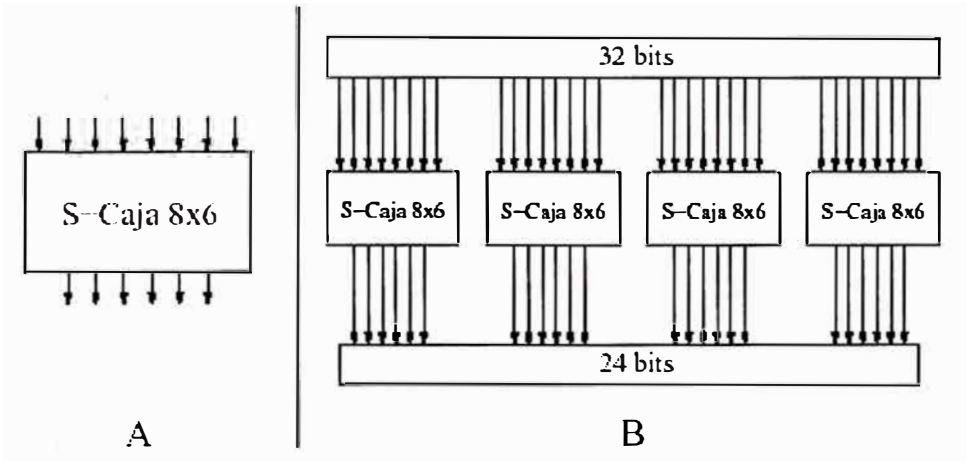

Figura 2.1. A: S-Caja individual. B: combinación de cuatro S-Cajas

Una S-Caja de m*n bits (ver Figura 2.1) es una tabla de sustitución que toma como entrada cadenas de m bits y da como salida cadenas de n bits. DES, por ejemplo, emplea ocho S-Cajas de 6*4 bits. La utilización de las S-Cajas es sencilla:

se divide el bloque original en trozos de m bits y cada uno de ellos se sustituye por otro de n bits, haciendo uso de la S-Caja correspondiente. Normalmente, cuanto más grandes sean las S-Cajas, más resistente será el algoritmo resultante, aunque la elección de los valores de salida para que den lugar a un buen algoritmo no es en absoluto trivial. Existe un algoritmo criptográfico, llamado CAST, que emplea seis S-Cajas de 8*32 bits. CAST codifica bloques de 64 bits empleando claves de 64 bits, consta de ocho rondas y deposita prácticamente toda su fuerza en las S-Cajas. De hecho, existen muchas variedades de CAST, cada una con sus S-Cajas correspondientes —algunas de ellas secretas—. Este algoritmo se ha demostrado resistente a las técnicas habituales de criptoanálisis, y sólo se conoce la fuerza bruta como mecanismo para atacarlo.

2.2.2. Algoritmo DES

Es el algoritmo simétrico más extendido mundialmente. Se basa en el algoritmo LUCIFER, que había sido desarrollado por IBM a principios de los setenta, y fue adoptado como estándar por el Gobierno de los EE.UU. para comunicaciones no clasificadas en 1976. En realidad la NSA lo diseñó para ser implementado por hardware, creyendo que los detalles iban a ser mantenidos en secreto, pero la Oficina Nacional de Estandarización publicó su especificación con suficiente detalle como para que cualquiera pudiera implementarlo por software. No fue casualidad que el siguiente algoritmo adoptado (Skipjack) fuera mantenido en secreto.

A mediados de 1998, se demostró que un ataque por la fuerza bruta a DES era viable, debido a la escasa longitud que emplea en su clave. No obstante, el algoritmo

aún no ha demostrado ninguna debilidad grave desde el punto de vista teórico, por lo que su estudio sigue siendo plenamente interesante.

El algoritmo DES codifica bloques de 64 bits empleando claves de 56 bits. Es una Red de Feistel de 16 rondas, más dos permutaciones, una que se aplica al principio (P_i) y otra que se aplica al final (P_f), tales que $P_i = P_f^{-1}$.

2.2.2.1. Claves débiles DES

El algoritmo DES presenta algunas claves débiles. En general, todos aquellos valores de la llave que conducen a una secuencia inadecuada de K_i serán poco recomendables.

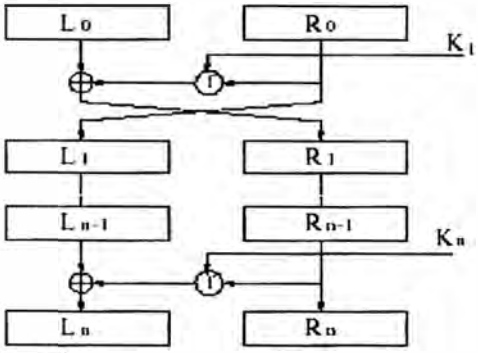

Figura 2.2. Estructura de una red de Feistel

Distinguiremos entre claves débiles, que son aquellas que generan un conjunto de dieciséis valores iguales de K_i —y que cumplen $E_k(E_k(M)) = M$, y claves semidébiles (ver Figura 2.2), que generan dos valores diferentes de K_i, cada uno de los cuales aparece ocho veces. En cualquier caso, el número de llaves de este tipo es tan pequeño en comparación con el número total de posibles claves.

2.2.3. Variantes DES

A mediados de Julio de 1998, una empresa sin ánimo de lucro, llamada EFF (Electronic Frontier Foundation), logró fabricar una máquina capaz de descifrar un

mensaje DES en menos de tres días. Curiosamente, pocas semanas antes, un alto cargo de la NSA había declarado que dicho algoritmo seguía siendo seguro, y que descifrar un mensaje resultaba aún excesivamente costoso, incluso para organizaciones gubernamentales.

A pesar de su caída, DES sigue siendo ampliamente utilizado en multitud de aplicaciones, como por ejemplo las transacciones de los cajeros automáticos. De todas formas, el problema real de DES no radica en su diseño, sino en que emplea una clave demasiado corta (56 bits), lo cual hace que con el avance actual de las computadoras los ataques por la fuerza bruta comiencen a ser opciones realistas.

Mucha gente se resiste a abandonar este algoritmo, precisamente porque ha sido capaz de sobrevivir durante veinte años sin mostrar ninguna debilidad en su diseño.

2.2.3.1. DES Múltiples

Consiste en aplicar varias veces el algoritmo DES con diferentes claves al mensaje original. Se puede hacer ya que DES no presenta estructura de grupo. El más común de todos ellos es el Triple-DES, que responde a la siguiente estructura:

$$C = E_{k1}(E^{-1}{}_{k2}\,(E_{k1}(M))) \tag{2.5}$$

Es decir, codificamos con la subclave k_1, decodificamos con k_2 y volvemos a codificar con k_1. La clave resultante es la concatenación de k_1 y k_2, con una longitud de 112 bits.

2.2.3.2. DES Generalizado

Esta variante emplea n trozos de 32 bits en cada ronda en lugar de dos, por lo que aumentamos tanto la longitud de la clave como el tamaño de mensaje que

se puede codificar, manteniendo sin embargo el orden de complejidad del algoritmo.

Se ha demostrado sin embargo que no sólo se gana poco en seguridad, sino que en muchos casos incluso se pierde.

2.2.3.3. DES con S-Cajas alternativas

Consiste en utilizar S-Cajas diferentes a las de la versión original de DES. En la práctica no se han encontrado S-Cajas mejores que las propias de DES. De hecho, algunos estudios han revelado que las S-Cajas originales presentan propiedades que las hacen resistentes a técnicas de criptoanálisis que no fueron conocidas fuera de la NSA hasta muchos años después de la aparición del algoritmo.

2.2.4. Algoritmo IDEA

El algoritmo IDEA (International Data Encryption Algorithm) es bastante más joven que DES, pues data de 1992. Para muchos constituye el mejor y más seguro algoritmo simétrico disponible en la actualidad. Trabaja con bloques de 64 bits de longitud y emplea una clave de 128 bits. Como en el caso de DES, se usa el mismo algoritmo tanto para cifrar como para descifrar.

IDEA es un algoritmo bastante seguro, y hasta ahora se ha mostrado resistente a multitud de ataques, entre ellos el criptoanálisis diferencial. No presenta claves débiles, y su longitud de clave hace imposible en la práctica un ataque por la fuerza bruta. Como ocurre con todos los algoritmos simétricos de cifrado por bloques, IDEA se basa en los conceptos de confusión y difusión, haciendo uso de las siguientes operaciones elementales (todas ellas fáciles de implementar):

a) XOR.

b) Suma módulo 2^{16}.

c) Producto módulo $2^{16} + 1$.

El algoritmo IDEA consta de ocho rondas. Dividiremos el bloque X a codificar, de 64 bits, en cuatro partes X_1, X_2, X_3 y X_4 de 16 bits. Para la interpretación entera de dichos registros se empleará el criterio *big endian*, lo cual significa que el primer *byte* es el más significativo. Denominaremos Z_i a cada una de las 52 subclaves de 16 bits que vamos a necesitar. Las operaciones de rondas son:

a) Multiplicar X_1 por Z_1.

b) Sumar X_2 con Z_2.

c) Sumar X_3 con Z_3.

d) Multiplicar X_4 por Z_4.

e) Hacer un XOR entre los resultados del paso 1 y el paso 3.

f) Hacer un XOR entre los resultados del paso 2 y el paso 4.

g) Multiplicar el resultado del paso 5 por Z_5.

h) Sumar los resultados de los pasos 6 y 7.

i) Multiplicar el resultado del paso 8 por Z_6.

j) Sumar los resultados de los pasos 7 y 9.

k) Hacer un XOR entre los resultados de los pasos 1 y 9.

l) Hacer un XOR entre los resultados de los pasos 3 y 9.

m) Hacer un XOR entre los resultados de los pasos 2 y 10.

n) Hacer un XOR entre los resultados de los pasos 4 y 10.

Las primeras ocho subclaves se calculan dividiendo la clave de entrada en bloques de 16 bits. Las siguientes ocho se calculan rotando la clave de entrada 25 bits a la izquierda y volviendo a dividirla, y así sucesivamente.

Las subclaves necesarias para descifrar se obtienen cambiando de orden las Z_i y calculando sus inversas para la suma o la multiplicación. Puesto que $2^{16} + 1$ es un número primo, nunca podremos obtener cero como producto de dos números, por lo que no necesitamos representar dicho valor. Cuando estemos calculando productos, utilizaremos el cero para expresar el número 216 —un uno seguido de 16 ceros—. Esta representación es coherente puesto que los registros que se emplean internamente en el algoritmo poseen únicamente 16 bits.

2.2.5. Algoritmo Rinjdael (AES)

En Octubre de 2000 el NIST (National Institute for Standards and Technology) anunciaba oficialmente la adopción del algoritmo Rijndael (pronunciado más o menos como reindal) como nuevo Estándar Avanzado de Cifrado (AES) para su empleo en aplicaciones criptográficas no militares, culminando así un proceso de más de tres años, encaminado a proporcionar a la comunidad internacional un nuevo algoritmo de cifrado potente, eficiente, y fácil de implementar. DES tenía por fin un sucesor.

La palabra Rijndael —en adelante, para referirnos a este algoritmo, emplearemos la denominación AES— es un acrónimo formado por los nombres de sus dos autores, los belgas Joan Ademen y Vincent Rijmen. Su interés radica en que todo el proceso de selección, revisión y estudio tanto de este algoritmo como de los restantes candidatos, se ha efectuado de forma pública y abierta, por lo que, prácticamente por primera vez, toda la comunidad criptográfica mundial ha participado en su análisis, lo cual convierte a Rijndael en un algoritmo perfectamente digno de la confianza de todos. AES es un sistema de cifrado por bloques, diseñado

para manejar longitudes de clave y de bloque variables, ambas comprendidas entre los 128 y los 256 bits. Realiza varias de sus operaciones internas a nivel de byte, interpretando éstos como elementos de un cuerpo de Galois GF (28). El resto de operaciones se efectúan en términos de registros de 32 bits. Sin embargo, en algunos casos, una secuencia de 32 bits se toma como un polinomio de grado inferior a 4, cuyos coeficientes son a su vez polinomios en GF (28).

Si bien, como ya se ha dicho, este algoritmo soporta diferentes tamaños de bloque y clave, en el estándar adoptado por el Gobierno Estadounidense en noviembre de 2001 (FIPS PUB 197), se especifica una longitud fija de bloque de 128 bits (Nb = 4) y la longitud de clave a escoger entre 128, 192 y 256 bits.

2.2.5.1. Estructura AES

AES, a diferencia de algoritmos como DES, no posee estructura de red de Feistel. En su lugar se ha definido cada ronda como una composición de cuatro funciones invertibles diferentes, formando tres capas, diseñadas para proporcionar resistencia frente a criptoanálisis lineal y diferencial. Cada una de las funciones tiene un propósito preciso:

La capa de mezcla lineal, funciones *DesplazarFila* y *MezclarColumnas*— permite obtener un alto nivel de difusión a lo largo de varias rondas.

La capa no lineal, función *ByteSub*— consiste en la aplicación paralela de s-cajas con propiedades óptimas de no linealidad. ***La capa de adición de clave,*** es un simple *or-exclusivo* entre el estado intermedio y la subclave correspondiente a cada ronda.

2.2.5.2. Elementos de AES

AES es un algoritmo que se basa en aplicar un número determinado de rondas a un valor intermedio que se denomina estado. Dicho estado puede

representarse mediante una matriz rectangular de bytes, que posee cuatro filas, y N_b columnas.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$

Figura 2.3. Ejemplo de matriz de estado con N_b=5 (160 bits)

Así, por ejemplo, si nuestro bloque tiene 160 bits (ver Figura 2.3), N_b será igual a 5. La llave tiene una estructura análoga a la del estado, y se representará mediante una tabla con cuatro filas y N_k columnas. Si nuestra clave tiene, por ejemplo, 128 bits, N_k será igual a 4 (ver Figura 2.4).

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Figura 2.4. Ejemplo de matriz de estado con N_k=4 (128 bits)

En algunos casos, tanto el estado como la clave se consideran como vectores de registros de 32 bits, estando cada registro constituido por los bytes de la columna correspondiente, ordenados de arriba a abajo.

El bloque que se pretende cifrar o descifrar se traslada directamente byte a byte sobre la matriz de estado, siguiendo la secuencia $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{0,1}$ . . ., y análogamente, los bytes de la clave se copian sobre la matriz de clave en el mismo orden, a saber, $k_{0,0}$, $k_{2,0}$, $k_{3,0}$, $k_{0,1}$ . . ..

Siendo B el bloque que queremos cifrar, y S la matriz de estado, el algoritmo AES con ***n*** rondas queda como sigue:

a) Calcular K0, K1,. . .Kn subclaves a partir de la clave K.

b) S ← B ⊕ K0

c) Para i = 1 hasta n hacer

d) Aplicar ronda i–ésima del algoritmo con la subclave K_i.

Puesto que cada ronda es una sucesión de funciones invertibles, el algoritmo de descifrado consistirá en aplicar las inversas de cada una de las funciones en el orden contrario, y utilizar los mismos K_i que en el cifrado, sólo que comenzando por el último.

2.2.5.3. Las rondas AES

Puesto que AES permite emplear diferentes longitudes tanto de bloque como de clave, el número de rondas requerido en cada caso es variable. En la Figura 2.5 se especifica cuántas rondas son necesarias en función de N_b y N_k.

	$N_b = 4$ (128 bits)	$N_b = 6$ (192 bits)	$N_b = 8$ (256 bits)
$N_k = 4$ (128 bits)	10	12	14
$N_k = 6$ (192 bits)	12	12	14
$N_k = 8$ (256 bits)	14	14	14

Figura 2.5. Ejemplo de matriz de estado con N_k=4 (128 bits), rondas AES

Siendo S la matriz de estado, y la subclave correspondiente a la ronda i–ésima, cada una de las rondas posee la siguiente estructura:

a) S ←ByteSub(S)

b) S ←DesplazarFila(S)

c) S ←MezclarColumnas(S)

d) S ← K_i ⊕ S

La última ronda es igual a las anteriores, pero eliminando el paso c.

2.2.5.4. La Seguridad AES

Según sus autores, es altamente improbable que existan claves débiles o semidébiles en AES, debido a la estructura de su diseño, que busca eliminar la simetría en las subclaves.

También se ha comprobado que es resistente a criptoanálisis tanto lineal como diferencial. En efecto, el método más eficiente conocido hasta la fecha para recuperar la clave a partir de un par texto cifrado-texto claro es la búsqueda exhaustiva, por lo que podemos considerar a este algoritmo uno de los más seguros en la actualidad.

CAPÍTULO III

CRIPTOGRAFIA DE LLAVE PÚBLICA

3.1. Algoritmos asimétricos de cifrado

Los algoritmos de llave pública, o algoritmos asimétricos, han demostrado su interés para ser empleados en redes de comunicación inseguras (Internet). Introducidos por Whitfield Diffie y Martin Hellman a mediados de los años 70, su novedad fundamental con respecto a la criptografía simétrica es que las claves no son únicas, sino que forman pares. Hasta la fecha han aparecido multitud de algoritmos asimétricos, la mayoría de los cuales son inseguros; otros son poco prácticos, bien sea porque el criptograma es considerablemente mayor que el mensaje original, bien sea porque la longitud de la clave es enorme. Se basan en general en plantear al atacante problemas matemáticos difíciles de resolver. En la práctica muy pocos algoritmos son realmente útiles. El más popular por su sencillez es RSA, que ha sobrevivido a multitud de ataques, si bien necesita una longitud de clave considerable. Otros algoritmos son los de El Gamal y Rabin.

Los algoritmos asimétricos emplean generalmente longitudes de clave mucho mayores que los simétricos. Por ejemplo, mientras que para algoritmos simétricos se considera segura una clave de 128 bits, para algoritmos asimétricos —si exceptuamos aquellos basados en curvas elípticas— se recomiendan claves de al

menos 1024 bits. Además, la complejidad de cálculo que comportan estos últimos los hace considerablemente más lentos que los algoritmos de cifrado simétricos. En la práctica los métodos asimétricos se emplean únicamente para codificar la clave de sesión (simétrica) de cada mensaje o transacción particular.

3.1.1. Aplicaciones de los algoritmos asimétricos

Los algoritmos asimétricos poseen dos claves diferentes en lugar de una, k_p y K_P, denominadas clave privada y clave pública. Una de ellas se emplea para codificar, mientras que la otra se usa para decodificar. Dependiendo de la aplicación que le demos al algoritmo, la clave pública será la de cifrado o viceversa. Para que estos criptosistemas sean seguros también ha de cumplirse que a partir de una de las claves resulte extremadamente difícil calcular la otra.

3.1.1.1. Protección de la información

Una de las aplicaciones inmediatas de los algoritmos asimétricos es el cifrado de la información sin tener que transmitir la clave de decodificación, lo cual permite su uso en canales inseguros.

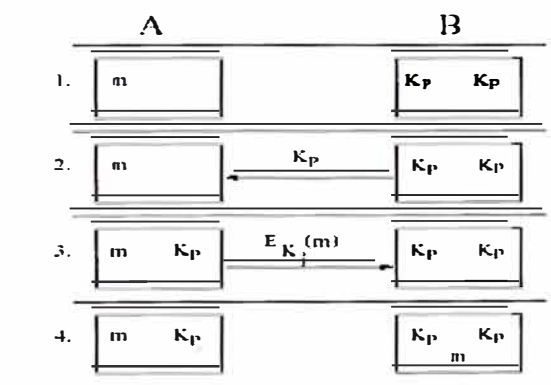

Figura 3.1. Transmisión de información empleando algoritmos asimétricos

Supongamos que A quiere enviar un mensaje a B (ver Figura 3.1). Para ello solicita a B su clave pública K_P. A genera entonces el mensaje cifrado EKP (m). Una

vez hecho esto únicamente quien posea la clave Kp —en nuestro ejemplo, B— podrá recuperar el mensaje original m. Nótese que para este tipo de aplicación, la llave que se hace pública es aquella que permite codificar los mensajes, mientras que la llave privada es aquella que permite descifrarlos.

3.1.1.2. Autentificación

La segunda aplicación de los algoritmos asimétricos es la autentificación de mensajes, con ayuda de funciones resumen, que nos permiten obtener una firma digital a partir de un mensaje. Dicha firma es mucho más pequeña que el mensaje original, y es muy difícil encontrar otro mensaje que dé lugar a la misma. Supongamos que A recibe un mensaje m de B y quiere comprobar su autenticidad. Para ello B genera un resumen del mensaje r(m) (ver Figura 3.2) y lo codifica empleando la clave de cifrado, que en este caso será privada.

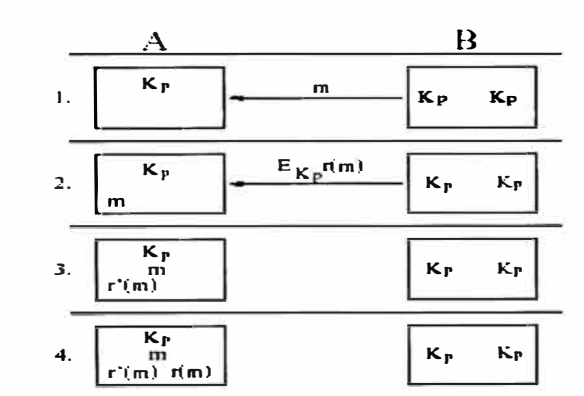

Figura 3.2. Autentificación de información empleando algoritmos asimétricos

La clave de descifrado se habrá hecho pública previamente, y debe estar en poder de A. B envía entonces a A el criptograma correspondiente a r (m). A puede ahora generar su propia r0 (m) y compararla con el valor r (m) obtenido del criptograma enviado por B. Si coinciden, el mensaje será auténtico, puesto que el único que posee la clave para codificar es precisamente B. Nótese que en este caso la

clave que se emplea para cifrar es la clave privada, justo al revés que para la simple codificación de mensajes.

En muchos de los algoritmos asimétricos ambas claves sirven tanto para cifrar como para descifrar, de manera que si empleamos una para codificar, la otra permitirá decodificar y viceversa. Esto ocurre con el algoritmo RSA, en el que un único par de claves es suficiente para codificar y autentificar.

3.1.2. El Algoritmo RSA

De entre todos los algoritmos asimétricos, quizá RSA sea el más sencillo de comprender e implementar. Como ya se ha dicho, sus claves sirven indistintamente tanto para codificar como para autentificar. Debe su nombre a sus tres inventores: Ronald Rivest, Adi Shamir y Leonard Adleman, y estuvo bajo patente de los Laboratorios RSA hasta el 20 de septiembre de 2000, por lo que su uso comercial estuvo restringido hasta esa fecha. De hecho, las primeras versiones de PGP lo incorporaban como método de cifrado y firma digital, pero se desaconsejó su uso a partir de la versión 5 en favor de otros algoritmos, que por entonces sí eran libres. Sujeto a múltiples controversias, desde su nacimiento nadie ha conseguido probar o rebatir su seguridad, pero se le tiene como uno de los algoritmos asimétricos más seguros. RSA se basa en la dificultad para factorizar grandes números. Las claves pública y privada se calculan a partir de un número que se obtiene como producto de dos primos grandes. El atacante se enfrentará, si quiere recuperar un texto claro a partir del criptograma y la llave pública, a un problema de factorización. Para generar un par de llaves (K_P,k_p), en primer lugar se eligen aleatoriamente dos números primos grandes, p y q. Después se calcula el producto n = pq.

3.1.2.1. Seguridad del algoritmo RSA

Técnicamente no es del todo cierto que el algoritmo RSA deposite su fuerza en el problema de la factorización. En realidad el hecho de tener que factorizar un número para descifrar un mensaje sin la clave privada es una mera conjetura. Nadie ha demostrado que no pueda surgir un método en el futuro que permita descifrar un mensaje sin usar la clave privada y sin factorizar el módulo n. De todas formas, este método podría ser empleado como una nueva técnica para factorizar números enteros, por lo que la anterior afirmación se considera en la práctica cierta. De hecho, existen estudios que demuestran que incluso recuperar sólo algunos bits del mensaje original resulta tan difícil como descifrar el mensaje entero. Aparte de factorizar n, podríamos intentar calcular $\Phi(n)$ directamente, o probar por la fuerza bruta tratando de encontrar la clave privada. Ambos ataques son más costosos computacional mente que la propia factorización de n, afortunadamente. Otro punto que cabría preguntarse es qué pasaría si los primos p y q que escogemos realmente fueran compuestos. Recordemos que los algoritmos de prueba de primos que conocemos son probabilísticos, por lo que jamás tendremos la absoluta seguridad de que p y q son realmente primos. Pero obsérvese que si aplicamos, por ejemplo, treinta pasadas del algoritmo de Rabin-Miller, las probabilidades de que el número escogido pase el test y siga siendo primo son de una contra 2^{60}: resulta más fácil que nos toque la primitiva y que simultáneamente nos parta un rayo. Por otra parte, si p o q fueran compuestos, el algoritmo RSA simplemente no funcionaría correctamente.

3.1.2.2. Vulnerabilidades del algoritmo RSA

Aunque el algoritmo RSA es bastante seguro conceptualmente, existen algunos puntos débiles en la forma de utilizarlo que pueden ser aprovechados por un

atacante. En esta sección comentaremos estas posibles vulnerabilidades, así como la forma de evitar que surjan.

Claves Débiles en RSA, se puede demostrar matemáticamente que existen ciertos casos para los cuales el algoritmo RSA deja el mensaje original tal cual, es decir

$$m^e = m \pmod{n} \tag{2.6}$$

En realidad, siempre hay mensajes que quedan inalterados al ser codificados mediante RSA, sea cual sea el valor de n. Nuestro objetivo será reducir al mínimo el número de éstos. Se puede comprobar que, siendo n = pq y e el exponente para codificar.

Claves Demasiado Cortas, actualmente se considera segura una clave RSA con una longitud de n de al menos 768 bits, si bien se recomienda el uso de claves no inferiores a 1024 bits. Hasta hace relativamente poco se recomendaban 512 bits, pero en mayo de 1999, Adi Shamir presentó el denominado dispositivo Twinkle, un ingenio capaz de factorizar números de manera muy rápida, aprovechando los últimos avances en la optimización de algoritmos específicos para esta tarea. Este dispositivo, aún no construido, podría ser incorporado en ordenadores de bajo coste y pondría en serio peligro los mensajes cifrados con claves de 512 bits o menos. Teniendo en cuenta los avances de la tecnología, y suponiendo que el algoritmo RSA no sea roto analíticamente, deberemos escoger la longitud de la clave en función del tiempo que queramos que nuestra información permanezca en secreto. Efectivamente, una clave de 1024 bits parece a todas luces demasiado corta como para proteger información por más de unos pocos años.

Ataques de Intermediario, el ataque de intermediario (ver Figura 3.3) puede darse con cualquier algoritmo asimétrico. Supongamos que A quiere establecer una

comunicación con B, y que C quiere espiarla. Cuando A le solicite a B su clave pública K_B, C se interpone, obteniendo la clave de B y enviando a A una clave falsa kC creada por él. Cuando A codifique el mensaje, C lo interceptará de nuevo, decodificándolo con su clave propia y empleando K_B para recodificarlo y enviarlo a B. Ni A ni B son conscientes de que sus mensajes están siendo interceptados.

La única manera de evitar esto consiste en asegurar a A que la clave pública que tiene de B es auténtica. Para ello nada mejor que ésta esté firmada por un amigo común, que certifique la autenticidad de la clave. En la actualidad existen los llamados anillos de confianza, que permiten certificar la autenticidad de las claves sin necesidad de centralizar el proceso. Por eso se nos recomienda cuando instalamos paquetes como el PGP que firmemos todas las claves sobre las que tengamos certeza de su autenticidad, y solamente esas.

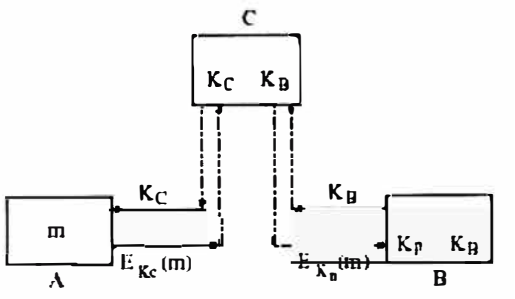

Figura 3.3. Ataque de intermediario para un algoritmo asimétrico

3.1.3. Otros algoritmos asimétricos

3.1.3.1. Algoritmo Diffie-Hellman

Es un algoritmo asimétrico, basado en el problema de Diffie-Hellman, que se emplea fundamentalmente para acordar una clave común entre dos interlocutores, a través de un canal de comunicación inseguro. La ventaja de este

sistema es que no son necesarias llaves públicas en el sentido estricto, sino una información compartida por los dos comunicantes.

Sean A y B los interlocutores en cuestión. En primer lugar, se calcula un número primo p y un generador α de Zp, con $2 \leq \alpha \leq p-2$. Esta información es pública y conocida por ambos. El algoritmo queda como sigue:

1) A escoge un número aleatorio x, comprendido entre 1 y p - 2 y envía a B el valor α^x (mód p).

2) B escoge un número aleatorio y, análogamente al paso anterior, y envía a A el valor α^y (mód p).

3) B recoge α^x y calcula $K = (\alpha^x)^y$ (mód p).

4) A recoge α^y y calcula $K = (\alpha^y)^x$ (mód p).

Puesto que x e y no viajan por la red, al final A y B acaban compartiendo el valor de K, sin que nadie que capture los mensajes transmitidos pueda repetir el cálculo.

3.1.3.2. Algoritmo El Gamal

Fue diseñado en un principio para producir firmas digitales, pero posteriormente se extendió también para codificar mensajes. Se basa en el problema de los logaritmos discretos, que está íntimamente relacionado con el de la factorización, y en el de Diffie-Hellman. Para generar un par de llaves, se escoge un número primo n y dos números aleatorios p y x menores que n. Se calcula entonces

$$y = p^x \text{ (mód n)} \qquad (2.7)$$

La llave pública es (p, y, n), mientras que la llave privada es x. Escogiendo n primo, garantizamos que sea cual sea el valor de p, el conjunto $\{p, p^2, p^3 \ldots\}$ es una permutación del conjunto {1, 2,..., n - 1}. Nótese que esto no es necesario para que el

algoritmo funcione, por lo que podemos emplear realmente un valor de n no primo, siempre que el conjunto generado por las potencias de p sea lo suficientemente grande.

3.1.3.3. Algoritmo de Rabin

El sistema de llave asimétrica de Rabin se basa en el problema de calcular raíces cuadradas módulo un número compuesto. Este problema se ha demostrado que es equivalente al de la factorización de dicho número.

En primer lugar escogemos dos números primos, p y q, ambos congruentes con 3 módulo 4 (los dos últimos bits a 1). Estos primos son la clave privada. La clave pública es su producto, $n = pq$.

Para codificar un mensaje m, simplemente se calcula $c = m^2$ (mód n).

3.1.3.4. Algoritmo DSA

El algoritmo DSA (Digital Signature Algorithm) es una parte el estándar de firma digital DSS (Digital Signature Standard). Este algoritmo, propuesto por el NIST, data de 1991, es una variante del método asimétrico de El Gamal.

Creación del par de llaves pública-llave-privada; el algoritmo de generación de claves es el siguiente:

a) Seleccionar un número primo q tal que $2^{159} < q < 2^{160}$.

b) Escoger t tal que $0 \leq t \leq 8$, y seleccionar un número primo p tal que $2^{511+64t} < p < 2^{512+64t}$, y que además q sea divisor de (p - 1).

c) Seleccionar un elemento $g \in Z_p$ y calcular $\alpha = g^{(p-1)/q}$ mód p.

d) Si $\alpha = 1$ volver al paso c

e) Seleccionar un número entero aleatorio a, tal que $1 \leq a \leq q - 1$

f) Calcular $y = \alpha^a$ mód p.

g) La clave pública es (p, q, α, y). La clave privada es a.

Generación y verificación de la firma; siendo h la salida de una función resumen sobre el mensaje m, la generación de una firma se hace mediante el siguiente algoritmo:

a) Seleccionar un número aleatorio k tal que $0 < k < q$.

b) Calcular $r = (\alpha^k$ mód p) mód q.

c) Calcular k^{-1} mód q.

d) Calcular $s = k^{-1} (h + ar)$ mód q.

e) La firma del mensaje m es el par (r, s).

El destinatario efectuará las siguientes operaciones, suponiendo que conoce la clave pública (p, q, α, y), para verificar la autenticidad de la firma:

a) Verificar que $0 < r < q$ y $0 < s < q$. En caso contrario, rechazar la firma.

b) Calcular el valor de h a partir de m.

c) Calcular $w = s^{-1}$ mód q.

d) Calcular $u1 = w \cdot h$ mód q y $u2 = w \cdot r$ mód q.

e) Calcular $v = (\alpha^{u1} y^{u2}$ mód p) mód q.

f) Aceptar la firma si y sólo si v = r.

3.1.4. Criptografía de curva elíptica

Para las curvas elípticas existe un problema análogo al de los logaritmos discretos en grupos finitos de enteros. Esto nos va a permitir trasladar cualquier algoritmo criptográfico definido sobre enteros, y que se apoye en este problema, al

ámbito de las curvas elípticas. La ventaja que se obtiene es que, con claves más pequeñas, se obtiene un nivel de seguridad equiparable.

Debido a la relación existente entre ambos, muchos algoritmos que se apoyan en el problema de la factorización pueden ser replanteados para descansar sobre los logaritmos discretos.

De hecho, existen versiones de curva elíptica de muchos de los algoritmos asimétricos más populares. A modo de ejemplo, en esta sección veremos cómo se redefine el algoritmo de cifrado de El Gamal.

3.1.4.1. Cifrado del El Gamal sobre curvas elípticas

Sea un grupo de curva elíptica, definido en GF(n) ó GF (2^n). Sea p un punto de la curva. Sea el conjunto (p), de cardinal n. Escogemos entonces un valor entero x comprendido entre 1 y n - 1, y calculamos

$$y = xp \tag{2.8}$$

La clave pública vendrá dada por (p, y, n), y la clave privada será x. El cifrado se hará escogiendo un número aleatorio k primo relativo con n. Seguidamente calculamos las expresiones

$$a = kp \tag{2.9}$$

$$b = m + ky \tag{2.10}$$

Siendo m el mensaje original representado como un punto de la curva. El criptograma será el par (a, b). Para descifrar, será suficiente con calcular

$$m = -(xa) + b \tag{2.11}$$

3.1.5. Los Protocolos SSL y TLS

El protocolo SSL (Secure Sockets Layer), desarrollado originalmente por la empresa Netscape, permite establecer conexiones seguras a través de Internet, de forma sencilla y transparente. Su fundamento consiste en interponer una fase de codificación de los mensajes antes de enviarlos por la red. Una vez que se ha establecido la comunicación, cuando una aplicación quiere enviar información a otra computadora, la capa SSL la recoge y la codifica, para luego enviarla a su destino a través de la red. Análogamente, el módulo SSL del otro ordenador se encarga de decodificar los mensajes y se los pasa como texto claro a la aplicación destinataria.
TLS (descrito en el documento RFC 2246) es un nuevo protocolo muy similar a SSL, ya que de hecho se basa en la versión 3.0 de este último, mejorándolo en algunos aspectos. Si bien su nivel de implantación aún no es muy elevado, todo parece indicar que está llamado a ser su sustituto. Una comunicación SSL o TLS consta fundamentalmente de dos fases.

Fase de saludo (handshaking): Consiste básicamente en una identificación mutua de los interlocutores, para la cual se emplean habitualmente los certificados X.509. Tras el intercambio de claves públicas, los dos sistemas escogen una clave de sesión, de tipo simétrico.

Fase de comunicación: En esta fase se produce el auténtico intercambio de información, que se codifica mediante la clave de sesión acordada en la fase de saludo.

Cada sesión lleva asociado un identificador único que evita la posibilidad de que un atacante escuche la red y repita exactamente lo mismo que ha oído, aún sin saber lo que significa, para engañar a uno de los interlocutores. Las ventajas de SSL

—y en el futuro, de TLS— son evidentes, ya que liberan a las aplicaciones de llevar a cabo las operaciones criptográficas antes de enviar la información, y su transparencia permite usarlo de manera inmediata sin modificar apenas los programas ya existentes. Desde hace tiempo los principales navegadores de Internet incorporan un módulo SSL, que se activa de forma automática cuando es necesario. Hasta diciembre de 1999, debido a las restricciones de exportación de material criptográfico existentes en los EE.UU., la mayoría de los navegadores incorporaban un nivel de seguridad bastante pobre (claves simétricas de 40 bits), por lo que conviene comprobar qué nivel de seguridad soporta nuestro navegador, y actualizarlo si fuera necesario.

3.2. Métodos de autentificación

Por autentificación entenderemos cualquier método que nos permita comprobar de manera segura alguna característica sobre un objeto. Dicha característica puede ser su origen, su integridad, su identidad, etc. Consideraremos tres grandes tipos dentro de los métodos de autentificación.

Autentificación de mensaje, queremos garantizar la procedencia de un mensaje conocido, de forma que podamos asegurarnos de que no es una falsificación. Este mecanismo se conoce habitualmente como firma digital.

Autentificación de usuario mediante contraseña, en este caso se trata de garantizar la presencia de un usuario legal en el sistema. El usuario deberá poseer una contraseña secreta que le permita identificarse.

Autentificación de dispositivo, se trata de garantizar la presencia de un dispositivo válido. Este dispositivo puede estar solo o tratarse de una llave electrónica que

sustituye a la contraseña para identificar a un usuario. Nótese que la autentificación de usuario por medio de alguna característica biométrica, como pueden ser las huellas digitales, la retina, el iris, la voz, etc. puede reducirse a un problema de autentificación de dispositivo, solo que el dispositivo en este caso es el propio usuario. De todas formas, en este capítulo únicamente trataremos métodos de autentificación basados en técnicas criptográficas.

3.2.1. Firmas digitales. Funciones resumen

La criptografía asimétrica permite autentificar información, es decir, poder asegurar que un mensaje m proviene de un emisor A y no de cualquier otro.

Asimismo vimos que la autentificación debía hacerse empleando una función resumen y no codificando el mensaje completo.

Sabemos que un mensaje *m* puede ser autentificado codificando con la llave privada k_p el resultado de aplicarle una función resumen, $E_{Kp}(r(m))$. Esa información adicional (que denominaremos *firma* o *signatura* del mensaje *m*) sólo puede ser generada por el poseedor de la clave privada k_p. Cualquiera que tenga la llave pública correspondiente estará en condiciones de decodificar y verificar la firma. Para que sea segura, la función resumen r(x) debe cumplir además ciertas características:

- r(m) es de longitud fija, independientemente de la longitud de m.
- Dado m, es fácil calcular r(m).
- Dado r(m), es computacional mente intratable recuperar m.
- Dado m, es computacional mente intratable obtener un m' / r(m) = r(m').

3.2.1.1. Longitud adecuada para una signatura

Para decidir cuál debe ser la longitud apropiada de una signatura, veamos primero el siguiente ejemplo: ¿Cuál es la cantidad *n* de personas que hay que poner en una habitación para que la probabilidad P de que el cumpleaños de una de ellas sea el mismo día que el mío supere el 50%? Sabemos que cuando n = 1, P = 1/365. Cuando n = 2, la probabilidad de que ningún cumpleaños coincida con el nuestro es el producto de la probabilidad de que no coincida el primero, por la probabilidad de que no coincida el segundo, luego:

$$P = 1 - (364/365).(364/365) \qquad (2.11)$$

En el caso general,

$$P = 1 - (364/365)^n \qquad (2.12)$$

Para que P > 0,5, n debe ser al menos igual a 253. Sin embargo, ¿cuál sería la cantidad de gente necesaria para que la probabilidad Q de que dos personas cualesquiera tengan el mismo cumpleaños supere el 50 %? Las dos primeras personas (o sea, cuando n = 2) tienen una probabilidad 364/365 de no compartir el cumpleaños; una tercera, supuesto que las dos primeras no lo comparten, tiene una probabilidad 363/365 de no compartirlo con las otras dos, por lo que tenemos (364/365) x (364/363), y así sucesivamente. En el caso general nos queda

$$Q = 1 - (364x363x362...x(365 - n + 1)/(365^n) \text{ con } 2 \leq n.$$

Si hacemos los cálculos, veremos que Q > 0,5 si n > 22, una cantidad sorprendentemente mucho menor que 253.

La consecuencia de este ejemplo, conocido como la paradoja del cumpleaños, es que aunque resulte muy difícil dado *m* calcular un m' tal que r(m) = r(m'), es considerablemente menos costoso generar muchos valores aleatoriamente, y posteriormente buscar entre ellos una pareja cualquiera (m,m'), tal que r(m) = r(m'). En el caso de una firma de 64 bits, necesitaríamos 2^{64} mensajes dado un *m* para obtener el m', pero bastaría con generar aproximadamente 232 mensajes aleatorios para que aparecieran dos con la misma signatura —en general, si la primera cantidad es muy grande, la segunda cantidad es aproximadamente su raíz cuadrada—. El primer ataque nos llevaría 600.000 años con una computadora que generara un millón de mensajes por segundo, mientras que el segundo necesitaría apenas una hora. Hemos de añadir pues a nuestra lista de condiciones sobre las funciones resumen la siguiente:

Debe ser difícil encontrar dos mensajes aleatorios, m y m', tales que r(m) = r(m').

Hoy por hoy se recomienda emplear signaturas de al menos 128 bits, siendo 160 bits el valor más usado.

3.2.1.2. Estructura de una función resumen

En general, las funciones resumen se basan en la idea de funciones de compresión, que dan como resultado bloques de longitud n a partir de bloques de longitud m. Estas funciones se encadenan de forma iterativa, haciendo que la entrada en el paso *i* sea función del *i-ésimo* bloque del mensaje y de la salida del paso *i - 1* (ver Figura 3.4). En general, se suele incluir en alguno de los bloques del mensaje *m* —al principio o al final—, información sobre la longitud total del mensaje. De esta forma se reducen las probabilidades de que dos mensajes con diferentes longitudes den el mismo valor en su resumen.

Figura 3.4. Estructura iterativa de una función resumen

3.2.1.3. Algoritmo MD5

Se trata de uno de los más populares algoritmos de generación de signaturas, debido en gran parte a su inclusión en las primeras versiones de PGP. Resultado de una serie de mejoras sobre el algoritmo MD4, diseñado por Ron Rivest, procesa los mensajes de entrada en bloques de 512 bits, y produce una salida de 128 bits. Siendo m un mensaje de b bits de longitud, en primer lugar se alarga *m* hasta que su longitud sea exactamente 64 bits inferior a un múltiplo de 512. El alargamiento se lleva a cabo añadiendo un 1 seguido de tantos ceros como sea necesario. En segundo lugar, se añaden 64 bits con el valor de *b*, empezando por el byte menos significativo. De esta forma tenemos el mensaje como un número entero de bloques de 512 bits, y además le hemos añadido información sobre su longitud.

En los últimos tiempos el algoritmo MD5 ha mostrado ciertas debilidades, aunque sin implicaciones prácticas reales, por lo que se sigue considerando en la actualidad un algoritmo seguro, si bien su uso tiende a disminuir.

3.2.1.4. Algoritmo SHA-1.

El algoritmo SHA-1 fue desarrollado por la NSA, para ser incluido en el estándar DSS (Digital Signature Standard). Al contrario que los algoritmos de cifrado propuestos por esta organización, SHA-1 se considera seguro y libre de puertas traseras, ya que el hecho de que el algoritmo sea realmente seguro favorece a

los propios intereses de la NSA. Produce firmas de 160 bits, a partir de bloques de 512 bits del mensaje original.

El algoritmo es similar a MD5, con la diferencia de que usa la ordenación *big endian*. Se inicializa de igual manera, es decir, añadiendo al final del mensaje un uno seguido de tantos ceros como sea necesario hasta completar 448 bits en el último bloque, para luego yuxtaponer la longitud en bits del propio mensaje —en este caso, el primer *byte* de la secuencia será el más significativo—. A diferencia de MD5, SHA-1 emplea cinco registros de 32 bits en lugar de cuatro.

3.2.1.5. Funciones de autentificación de mensaje

Frente a los MDC, vistos en la sección anterior, existe otra clase de funciones resumen, llamada genéricamente MAC (message authentication codes). Los MAC se caracterizan fundamentalmente por el empleo de una clave secreta para poder calcular la integridad del mensaje.

Puesto que dicha clave sólo es conocida por el emisor y el receptor, el efecto conseguido es que el receptor puede, mediante el cálculo de dicha función, comprobar tanto la integridad como la procedencia del mensaje.

Existen multitud de MAC diferentes, pero lo más común es cifrar el mensaje mediante un algoritmo simétrico en modo CBC(Cipher Book Chaining Mode), y emplear la salida correspondiente al cifrado del último bloque.

3.2.1.6. Criptoanálisis diferencial

Descubierto por Biham y Shamir en 1990, permite efectuar un ataque de texto claro escogido a DES que resulta más eficiente que la fuerza bruta. Se basa en el estudio de los pares de criptogramas que surgen cuando se codifican dos textos claros con diferencias particulares, analizando la evolución de dichas diferencias a lo

largo de las rondas de DES. Para llevar a cabo un criptoanálisis diferencial se toman dos mensajes cualesquiera (incluso aleatorios) idénticos salvo en un número concreto de bits. Usando las diferencias entre los textos cifrados, se asignan probabilidades a las diferentes claves de cifrado. Conforme tenemos más y más pares, una de las claves aparece como la más probable. Esa sería la clave buscada.

3.2.1.7. Criptoanálisis lineal

El criptoanálisis lineal, descubierto por Mitsuru Matsui, basa su funcionamiento en tomar algunos bits del texto claro y efectuar una operación XOR entre ellos, tomar algunos del texto cifrado y hacerles lo mismo, y finalmente hacer un XOR de los dos resultados anteriores, obteniendo un único bit. Efectuando esa operación a una gran cantidad de pares de texto claro y criptograma diferentes podemos ver si se obtienen más ceros o más unos. Si el algoritmo criptográfico en cuestión es vulnerable a este tipo de ataque, existirán combinaciones de bits que, bien escogidas, den lugar a un sesgo significativo en la medida anteriormente definida, es decir, que el número de ceros (o unos) es apreciablemente superior. Esta propiedad nos va a permitir poder asignar mayor probabilidad a unas claves sobre otras y de esta forma descubrir la clave que buscamos.

CAPÍTULO IV

IMPLEMENTACIONES DE REDES VPN

4.1. Seguridad en redes

La rápida expansión y popularización de Internet ha convertido a la seguridad en redes en uno de los tópicos más importantes dentro de la Informática moderna. Con tal nivel de interconexión, los virus y los *hackers* campan a sus anchas, aprovechando las deficientes medidas de seguridad tomadas por administradores y usuarios a los que esta nueva revolución ha cogido por sorpresa.

Las ventajas de las redes en Informática son evidentes, pero muchas veces se minusvaloran ciertos riesgos, circunstancia que a menudo pone en peligro la seguridad de los sistemas. En unos pocos años la inmensa mayoría de las empresas operarán a través de la Red, y esto sólo será posible si los profesionales de la Informática saben aportar soluciones que garanticen la seguridad de la información.

4.1.1. Importancia de las redes

La Informática es la ciencia del tratamiento automático de la información, pero tanto o más importante que su procesamiento y almacenamiento es la posibilidad de poder transmitirla de forma eficiente. La información tiene un tiempo de vida cada vez menor y la rapidez con la que pueda viajar es algo crucial. Los

últimos avances en compresión y transmisión de datos digitales permiten hoy por hoy transferir cantidades enormes de información a velocidades que hace tan solo unos años eran impensables. En este sentido las redes de computadoras desempeñan un papel fundamental en la Informática moderna. Pero hemos de tener en cuenta que la complejidad de las grandes redes y su carácter público convierte la protección física de los canales de comunicación en algo tremendamente difícil. Hemos de depositar nuestra confianza en la Criptografía para garantizar la confidencialidad en las comunicaciones. Uno de los mayores obstáculos que han tenido que ser superados para que las redes pudieran desarrollarse, ha sido encontrar lenguajes comunes para que computadoras de diferentes tipos pudieran entenderse. En este sentido el conjunto de protocolos TCP/IP se ha erigido como estándar de facto en la industria de la Informática. En general todas las redes de computadoras se construyen conceptualmente sobre diferentes capas de abstracción, que desarrollan tareas distintas y proporcionan un protocolo unificado a las capas superiores. La Criptografía podrá entonces ser empleada en diferentes niveles de abstracción. Por ejemplo, podemos cifrar un fichero antes de transmitirlo por la red, lo cual correspondería al nivel de abstracción mayor, o podemos enviarlo en claro, pero a través de un protocolo de bajo nivel que cifre cada uno de los paquetes de información en los que se va a subdividir el fichero en el momento de transmitirlo.

En función del tipo de red con el que trabajemos nos enfrentaremos a diferentes clases de riesgos, lo cual nos conducirá inevitablemente a medidas de diferente naturaleza para garantizar la seguridad en las comunicaciones. En este capítulo haremos una breve reflexión sobre algunos de los casos que pueden darse, sin tratar de ser exhaustivos —sería imposible, dada la inmensa cantidad de

posibilidades—. Nuestro objetivo se centrará en aportar una serie de directrices que nos permitan analizar cada situación y establecer una correcta política de protección de la información. Ya que no existe una solución universal para proteger una red, en la mayoría de los casos la mejor estrategia suele consistir en tratar de colarnos nosotros mismos para poner de manifiesto y corregir posteriormente los agujeros de seguridad que siempre encontraremos. Esta estrategia se emplea cada vez con mayor frecuencia, y en algunos casos hasta se contrata a hackers para que impartan cursillos de seguridad a los responsables de las redes de las empresas.

4.1.2. Redes internas

El caso más sencillo de red que nos podemos encontrar corresponde al término LAN(Local Area Network), de ámbito muy limitado geográficamente — usualmente un único edificio— con todos los computadores interconectados a través de unos cables de los que se es propietario. Esta última circunstancia nos va a permitir ejercer un control total sobre el canal de comunicaciones, pudiendo protegerlo físicamente, lo cual evita prácticamente cualquier peligro de falta de privacidad en la información.

Uno de los riesgos dignos de mención en estos casos son las posibles pérdidas de información debidas a fallos físicos, que pueden ser minimizados llevando a cabo una adecuada política de copias de respaldo, que deberán ser confeccionadas periódicamente, almacenadas en un lugar diferente de aquel donde se encuentra la red, y protegidas adecuadamente contra incendios y accesos no deseados.

Otro riesgo que se da en las redes locales, a menudo infravalorado, es el que viene del uso inadecuado del sistema por parte de los propios usuarios. Ya sea por mala fe

o descuido, un usuario con demasiados privilegios puede destruir información, por lo que estos permisos deben ser asignados con mucho cuidado por parte de los administradores. Esta circunstancia es muy importante, ya que, sobre todo en pequeñas empresas, el dueño muchas veces cree que debe conocer la clave del administrador, y luego es incapaz de resistir la tentación de jugar con ella, poniendo en serio peligro la integridad del sistema y entorpeciendo el trabajo del administrador o *superusuario*.

Existen redes internas en las que un control exhaustivo sobre el medio físico de transmisión de datos es en la práctica imposible. Piénsese en un edificio corporativo con un acceso no muy restringido, por ejemplo un aulario de una universidad, que posee conexiones ethernet en todas sus dependencias. En principio, nada impediría a una persona conectar un ordenador portátil a una de esas conexiones para llevar a cabo un análisis del tráfico de la red sin ser descubierta, o suplantar a cualquier otro computador. En estos casos será conveniente llevar a cabo algún tipo de control, como la deshabilitación dinámica de las conexiones de red no utilizadas en cada momento, la verificación del identificador único de la tarjeta de red concreta que debe estar conectada en cada punto, o la adopción de protocolos de autentificación de las computadoras dentro de la red, como por ejemplo Kerberos(protocolo que permite a un computador autentificarse dentro de una red). Uno de los últimos avances en redes locales son las denominadas WLAN(Gíreles Lan Area Network), muy en boga en la actualidad, que basan su funcionamiento en el empleo de ondas de radio como canal de comunicaciones, proporcionando una flexibilidad sin precedentes a las redes. Sin embargo, los protocolos propuestos hasta la fecha (IEEE 802.11 y 802.11b) proporcionan un nivel de seguridad realmente

deficiente, que podría permitir a un atacante analizar el tráfico con relativa facilidad, por lo que se recomienda emplear protocolos de cifrado en las capas de mayor nivel.

4.1.3. Redes externas

Consideraremos red externa a aquella que, en todo o en parte, se apoye en un canal físico de comunicación ajeno. En la actualidad, la mayor parte de las redes externas e internas están interconectadas, formando lo que conocemos como Internet. Existirán redes externas de muy diferentes tipos, pero todas ellas tienen en común la característica de que en algún momento la información viaja por canales sobre los que no se tiene ningún tipo de control. Todas las técnicas que nos van a permitir llevar a cabo protecciones efectivas de los datos deberán hacer uso necesariamente de la Criptografía.

Para identificar los posibles riesgos que presenta una red externa, hemos de fijarnos en cuestiones tan dispares como el sistema operativo que corre sobre los ordenadores o el tipo de acceso que los usuarios legales del sistema pueden llevar a cabo.

Una de las configuraciones más comunes consiste en el uso de una red local conectada al exterior mediante un *cortafuego* —computadora que filtra el tráfico entre la red interna y el exterior—. Los cortafuegos son herramientas muy poderosas si se emplean adecuadamente, pero pueden entrañar ciertos riesgos si se usan mal. Por ejemplo, existen muchos lugares donde el cortafuegos está conectado a la red local y ésta a su vez a la red externa (ver Figura 10, caso A). Esta configuración es la más sencilla y barata, puesto que sólo necesitamos una tarjeta de red en el cortafuego, pero no impediría a un computador situado en el exterior acceder

directamente a los de la red local. La configuración correcta se puede apreciar en el caso B de la Figura 4.1, donde la red externa (y todos sus peligros) está separada físicamente de la red local.

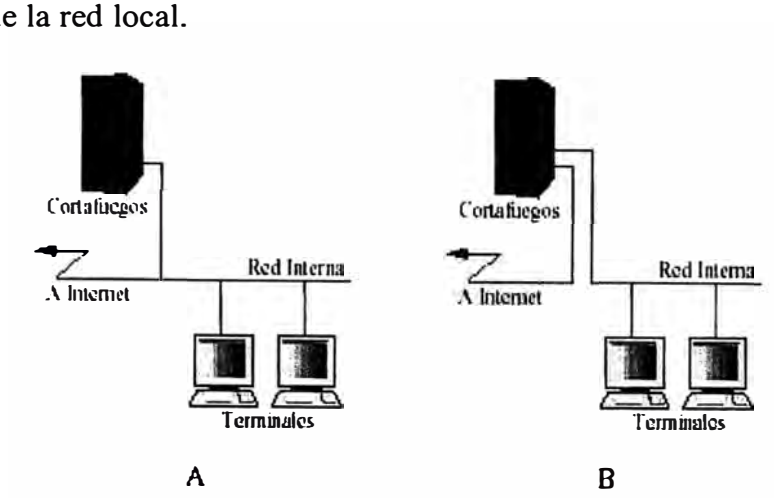

Figura 4.1. A: Configuración incorrecta, B: Configuración correcta

Podemos distinguir dos grandes tipos de peligros potenciales que pueden comprometer nuestra información desde una red externa

Ataques indiscriminados, suelen ser los más frecuentes, y también los menos dañinos. Dentro de esta categoría podemos incluir los troyanos y los virus, programas diseñados normalmente para colarse en cualquier sistema y producir efectos de lo más variopinto. Precisamente por su carácter general, existen programas específicos que nos protegen de ellos, como los antivirus. Conviene disponer de un buen antivir y actualizarlo periódicamente.

Ataques a medida, mucho menos comunes que los anteriores, y también más peligrosos, son los ataques que generalmente llevan a cabo los hackers. En estos casos las víctimas son casi siempre grandes corporaciones, y muchas veces la información ni siquiera es destruida o comprometida, puesto que los hackers sólo persiguen enfrentarse al reto que supone para ellos entrar en un sistema grande. El

problema es que para borrar sus huellas y dificultar el rastreo de sus acciones, suelen atacar en primer lugar sistemas pequeños para desde ellos cometer sus travesuras, lo cual convierte a cualquier sistema en potencial víctima de estos personajes. Lo que ocurre en la mayoría de los casos es que su necesidad de emplear sistemas pequeños como plataforma les obliga a no dañarlos, para no dejar ningún tipo de rastro que permita localizarlos posteriormente.

En cuanto a la protección de las comunicaciones en sí, baste decir que existen protocolos de comunicación segura de bajo nivel, como SSL y TLS, que permiten establecer comunicaciones seguras a través de Internet, haciendo uso de algoritmos simétricos y asimétricos simultáneamente. Estos protocolos son transparentes y pueden correr bajo otros ampliamente conocidos, como POP3, TELNET, FTP, HTTP, etc. De hecho, gran cantidad de aplicaciones los emplean en sus comunicaciones. Desgraciadamente, las restrictivas leyes norteamericanas en cuanto a la exportación de material criptográfico hacen que la gran mayoría de las aplicaciones seguras que se venden fuera de los EE.UU. y Canadá estén en realidad debilitadas, por lo que hemos de informarnos muy bien antes de depositar nuestra confianza en ellas.

4.2. Hackers

Se consideran a sí mismos una casta, y su filosofía de la vida es casi una religión. Delincuentes para unos, héroes para otros, multitud de leyendas circulan sobre estos personajes. Al igual que en otras épocas había alquimistas, capaces de dominar los oscuros poderes de la materia, hoy los *hackers* están considerados por muchos como los nigromantes de la era tecnológica. Nos guste o no, los hackers son

ya un mito en la cultura de finales del siglo XX. No debemos confundir el término *hacker* con el de pirata informático, ya que este último es un concepto más amplio. Aquellos que conozcan la cultura *cyberpunk*, sabrán que además de hackers hay otros grupos, tales como los crackers, que se dedican a la copia ilegal de software, y los phreakers, que dirigen sus esfuerzos hacia las compañías telefónicas. Cada uno de ellos se especializa en algún tipo de actividad curiosamente, las actuaciones de los crackers y phreakers suelen ser siempre delictivas, mientras que las de los hackers en algunos casos no lo son, mal que les pese a muchos. Por supuesto, todos ellos justifican sus formas de pensar y actuar con argumentos de lo más variopinto, que tienen como punto común la lucha contra el sistema establecido.

Periódicamente los medios de comunicación nos sorprenden con alguna nueva hazaña de estos personajes, contribuyendo, junto con la industria cinematográfica, al crecimiento y propagación de su leyenda.

4.2.1. El Hielo y los Vaqueros

Un *hacker* es un individuo que se dedica a infiltrarse en sistemas informáticos. Su actividad, tan antigua como las redes de ordenadores, conoce diversas variantes. Desde aquellos que no tratan de hacer ningún daño, y que consideran estas actuaciones como un excitante reto a su inteligencia, hasta aquellos cuyo único objetivo es sabotear una red, llevándose toda la información que posea para luego venderla, podemos decir que hay hackers para todos los gustos.

En 1983, William Gibson escribió "Neuromante", pieza clave de la literatura de Ciencia Ficción moderna y referencia obligada cuando se habla de cultura cyberpunk. En esta obra los vaqueros término empleado por Gibson para referirse a

los hackers burlaban el Hielo de los sistemas informáticos del inglés ICE, Intrusion Countermeasures Electronics a través del ciberespacio, también denominado matriz en el libro, o si se prefiere, matriz. Huelga decir que la magia y lo esotérico nada tienen que ver con estos sujetos. En general son individuos bastante ingeniosos y bien informados que se dedican a buscar y explotar fallos más o menos sutiles en los sistemas de seguridad. Puesto que cada sistema se puede decir que es único, los buenos vaqueros suelen elaborar ataques a medida, poniendo a prueba su profundo conocimiento sobre las redes de ordenadores. Es prácticamente imposible protegerse al cien por cien de un ataque de esta naturaleza, y debemos pensar que si, por alguna razón, nos convertimos en objetivo de un hacker lo suficientemente bueno, tarde o temprano acabaremos cayendo.

También hay que decir que aunque muchos actúan solos, los vaqueros suelen formar grupos, en los que cada uno tiene su alias, y que normalmente sólo establecen contacto a través de la Red, no conociéndose personalmente entre ellos. Suelen dominar bastante bien el uso de la Criptografía, y frecuentemente la emplean en sus comunicaciones. Alguno podría pensar que este es un buen argumento para imponer un control gubernamental sobre el uso privado de la Criptografía, pero en ese caso también deberíamos permitir que la policía entre en nuestras casas, lea nuestras cartas y escuche nuestras conversaciones telefónicas sin nuestro consentimiento, para evitar que cometamos delitos. Pero no todo va a ser negativo. La gente que instala un dispositivo antirrobo en su casa sabe que puede ser burlado por un ladrón altamente especializado, pero aún así lo considera seguro, ya que un ladrón profesional asumirá el riesgo sólo si la casa despierta el suficiente interés. Así pues, basta con tomar unas medidas de seguridad proporcionales al valor que posea el sistema que queremos

proteger. El problema es que en muchos casos, por simple desconocimiento, los sistemas están mal protegidos, hasta tal punto que es como si dejáramos abierta la puerta de nuestra casa. En esos casos, cualquiera puede entrar a fisgonear. Por desgracia, las computadoras son tan heterogéneas que no existe un conjunto de medidas universal que nos permita protegernos de estos chicos traviesos. Intentaremos no obstante dar unas pautas sobre las técnicas que más emplean para que así cada cual pueda saber qué medidas debe tomar en su caso concreto.

4.2.2. Cómo actúan los hackers?

Comentaremos en esta sección muy brevemente algunas de las técnicas más comunes empleadas para infiltrarse en computadores ajenos a través de la Red. Esto no quiere decir que sean las únicas técnicas posibles, ni siquiera que sean las mejores, pero servirán para hacernos una idea del modo de actuar de estos individuos.

Puerto	Función
21	FTP
23	Telnet
25	SMTP (Mail)
37	Time
43	Whois
80	HTTP (Servidor Web)
110	POP3 (Mail)
117	UUCP
119	NNTP (News)
513	Login
514	Shell
515	Spooler

Tabla 4.1. Algunos puertos TCP/IP

4.2.2.1. Protocolo TCP/IP. Demonios y Puertos

TCP/IP es el conjunto de protocolos que se ha impuesto como norma universal de facto en las comunicaciones. Internet se basa en dicho protocolo, y

aunque existan otros para redes locales, los sistemas operativos actuales permiten su coexistencia, por lo que podemos decir sin temor a equivocarnos que prácticamente cualquier máquina conectada a Internet entiende TCP/IP.

Un computador con TCP/IP puede establecer múltiples comunicaciones simultáneamente, a través de los denominados puertos. Un puerto se comporta como los canales de un televisor: a través de un único cable llegan muchas emisiones, de las cuales podemos escoger cuál ver con solo seleccionar el canal correspondiente. Existen puertos dedicados a tareas concretas (ver Tabla 2). Así por ejemplo el puerto 80 se emplea para las páginas Web, y el 21 para la transferencia de ficheros. En el cuadro 2 podemos ver algunos de los más usuales, aunque existen muchos más. Hay que decir que esta tabla es orientativa: nada nos impediría situar nuestro demonio de FTP en el puerto 300, por ejemplo, aunque eso obligaría a quienes quisieran establecer una comunicación FTP con nosotros a emplear dicho puerto. De hecho, ciertos servidores de acceso restringido emplean puertos no normalizados para evitar visitantes molestos.

Un demonio es un programa que se ejecuta y que escucha a través de un puerto a la espera de establecer comunicaciones. Así, por ejemplo, un servidor de páginas web tiene un demonio asociado al puerto 80, esperando solicitudes de conexión. Cuando nosotros cargamos una página en el navegador estamos enviando una solicitud al puerto 80 del servidor, que responde con la página correspondiente. Si el servidor web no estuviera ejecutando el demonio o éste estuviera escuchando en otro puerto, no podríamos consultar la página que buscamos.

Una vez que se establece la comunicación en un puerto, los ordenadores hablan entre ellos, usando diferentes idiomas, como por ejemplo HTTP para las

páginas Web, FTP para las transferencias de ficheros, etc. Ahora vamos a ver un ejemplo utilizando la orden telnet de UNIX y el protocolo SMTP de envío de correos electrónicos, que se ubica en el puerto 25:

```
usuario> telnet 1.2.3.4 25
Trying 1.2.3.4...
Connected to 1.2.3.4.
Escape character is '^]'.
220 host.dominio.pais ESMTP Sendmail 8.9.3; Fri, 10 Sep 1999 16:16:55
```

A partir de este momento el ordenador con IP 1.2.3.4 (host.dominio.pais) está esperando nuestros mensajes a través de su puerto 25. Si escribimos :

```
helo mlucena
```

el demonio responderá algo parecido a

```
250 host.dominio.pais Hello mlucena@host [3.2.5.6], pleased to meet you
```

Puesto que el demonio es un programa, puede que contenga errores, o que simplemente haya situaciones en las que no funcione adecuadamente. En los sistemas tipo UNIX, uno puede enviar mensajes extraños para los que el demonio no se encuentra preparado, y así lograr que aborte su ejecución, dejándonos una consola de texto con la que podremos tomar el control del sistema. En general, los fabricantes de software suelen actualizar periódicamente sus programas, subsanando paulatinamente los errores que éstos puedan contener, por lo que es de una importancia crucial que los demonios instalados en nuestro sistema sean fiables y se encuentren al día. Un fallo muy famoso y que dio mucho que hablar estaba en el demonio que escuchaba las comunicaciones SMB en las primeras versiones de Windows 95. Si uno enviaba un mensaje concreto a ese puerto podía bloquear de

forma instantánea el ordenador de su víctima, y para ello sólo tenía que conocer su número IP.

En general, el hacker se dedica a tratar de averiguar en qué puertos está escuchando el ordenador objetivo, y luego a localizar y explotar posibles fallos en los demonios correspondientes, para tomar el control del sistema. Muchas veces nuestro ordenador puede que esté escuchando algún puerto sin que nosotros lo sepamos. Existe un troyano que corre sobre los sistemas Windows, denominado Back Orifice, que escucha un puerto a la espera de que el ordenador atacante tome el control de nuestra máquina. Por desgracia, existen programas cuya configuración por defecto no es lo suficientemente conservadora, y que habilitan ciertas características a no ser que se les diga lo contrario, abriendo inevitablemente agujeros de seguridad. A modo de ejemplo, citaremos el problema que surgió en el verano de 1998, cuando se descubrió que era posible descargar, si el servidor corría bajo Windows NT, el código fuente de algunas páginas Web de tipo ASP de una manera muy simple, a no ser que el administrador tomara ciertas precauciones. Afortunadamente, el problema se solucionó con rapidez, y gracias a estos pequeños sustos cada vez se pone más cuidado a la hora de elaborar software sensible.

4.2.2.2. Desbordamiento de búffer

Uno de los mecanismos más empleados para hacer saltar a un demonio es el desbordamiento de *buffer*, que aprovecha una característica muy común en casi todos los compiladores de lenguajes de programación modernos. Cuando se ejecuta un procedimiento o subprograma, el ordenador reserva memoria para los datos de entrada al procedimiento, junto con información relativa a la dirección de memoria por la que debe continuar la ejecución una vez finalizado. En

algunos casos, los datos de entrada pueden ser de mayor tamaño que el espacio que el programador estimó suficiente en su día para albergarlos, por lo que, al salirse de su lugar, pueden sobrescribir y modificar otros campos. Un atacante podría entonces pasar al demonio una cadena que incluyera código ejecutable, y que machacara el campo que indica dónde debe seguir la ejecución del programa, para que apuntara precisamente a su código malicioso. De esta forma se podría ejecutar virtualmente cualquier cosa en la máquina de la víctima. Evidentemente, la construcción de cadenas de este tipo no es una tarea sencilla, pero una vez elaboradas, podríamos emplearlas cuantas veces quisiéramos, y desde cualquier otro ordenador.

4.2.2.3. Suplantando usuarios

Lo ideal para entrar en un sistema es hacerlo como administrador, lo cual proporciona suficientes privilegios como para alterar cualquier cosa sin ningún problema. A veces ocurre que el ordenador víctima no presenta vulnerabilidades en los puertos que escucha, por lo que debemos buscar otros medios para entrar en él. La mayoría de los sistemas operativos permiten la existencia de usuarios genéricos, llamados invitados, que no necesitan contraseña para entrar en el sistema y que tienen unos privilegios de acceso bastante limitados. En muchos casos esos privilegios pueden llegar a ser suficientes como para perpetrar un ataque con garantías de éxito, debido a que un invitado puede acceder al fichero que almacena las contraseñas. Afortunadamente, esta circunstancia ha sido subsanada en casi todos los sistemas operativos, pero ha de ser tenida en cuenta, para evitar desagradables sorpresas.

Si un usuario posee el fichero de contraseñas, puede llevar a cabo un ataque con diccionario, y eventualmente llegar a averiguar las claves de cada usuario, lo

cual le permitirá entrar en el sistema. Este tipo de ataque se volverá inútil si los usuarios escogen contraseñas adecuadas, del tipo que se propone en esta obra.

Una posibilidad bastante inquietante es la de, una vez que se han ganado suficientes privilegios, sustituir el fichero de contraseñas por otro elaborado por el hacker, lo cual dejaría sin acceso a todos los usuarios legítimos del sistema, ¡incluidos los administradores!. En tal caso habría que desconectar el sistema de la red y restaurarlo manualmente, con el consiguiente coste tanto de tiempo como de dinero.

4.2.2.4. Borrando huellas

Todos los sistemas operativos serios incorporan algún sistema de registro de los eventos del sistema que permite saber con detalle lo que en la computadora ha ido ocurriendo. Un vaquero que se precie debe eliminar todas las entradas de dicho registro relativas a su paso por el sistema, si no quiere que un policía llame a su puerta a los pocos días. . .

Además de tratar de borrar todas sus huellas, un hacker suele organizar sus ataques de forma que si queda algún rastro de su paso por el sistema elegido, éste sea realmente confuso. Para ello nada mejor que emplear otros ordenadores más modestos como plataforma para atacar al auténtico objetivo. Normalmente los vaqueros buscan ordenadores poco protegidos, entran en ellos, y controlándolos remotamente intentan encontrar las debilidades del objetivo real. Esta estrategia hará que en los registros del sistema atacado aparezcan datos sospechosos acerca del ordenador intermedio, pero pocas veces del auténtico enemigo.

4.2.2.5. Ataques pasivos y coordinados

Recientemente se han detectado fallos de seguridad en los navegadores de Internet que permitirían a un hipotético atacante colocar en su página

web código *malicioso*. De esta forma todos los que visitaran esa página y no hubieran tomado las adecuadas precauciones se verían afectados con problemas tales como la ejecución de algún programa posiblemente un virus, o el envío de algún fichero propio al atacante, todo ello de forma inadvertida, por supuesto. El atacante sólo tiene que esperar a que los incautos vayan cayendo. . .

Uno de los tipos de ataque que más fama ha cobrado últimamente es el que se conoce como Denegación de Servicio (en inglés Denial of Service, o DoS). éste se aprovecha de que los servidores, cuando reciben una solicitud de conexión, reservan memoria para atenderla. Basta, pues, con efectuar miles de solicitudes incompletas para bloquear la memoria del servidor, e impedir que acepte más conexiones. En la práctica, este ataque no permite robar información, sino únicamente paralizar un servidor, lo cual puede hacer suficiente daño en sitios Web que ofrecen servicios en línea, como los dedicados al comercio electrónico.

En realidad, la única modalidad efectiva de ataque DoS es aquella que se lleva a cabo, simultáneamente y de forma distribuida, por muchos computadores. El quid de la cuestión radica en que prácticamente cualquier tipo de ataque puede ser automatizado mediante un programa adecuado, por lo que un hacker podría construir un troyano que incorporara el código necesario para efectuar el ataque, y limitarse a coordinarlo cuando lo estime oportuno. Vulgarmente, se denomina zombie a aquella computadora que contiene un código capaz de contribuir en un ataque distribuido, de forma inadvertida para sus usuarios. Obsérvese que el crecimiento casi explosivo de Internet, impulsado en muchas ocasiones por políticas poco realistas de los gobiernos, está conduciendo a la existencia de un número cada vez mayor de computadoras con una deficiente administración, conectadas a la Red en bibliotecas,

colegios, etc, perfectas candidatas a protagonizar el próximo ataque distribuido, que posiblemente provocará pérdidas millonarias y hará correr ríos de tinta.

4.2.3. Cómo protegerse de los hackers?

Después de haber leído las anteriores secciones, parece una auténtica locura tener ordenadores conectados a Internet. Nada más lejos de la realidad. Así como hay muchas formas de poder entrar fraudulentamente en un ordenador, también hay muchas formas de protegerse razonablemente contra estos ataques. Vamos a dar una serie de consejos prácticos que, si no nos protegen totalmente, ponen las cosas bastante difíciles a los hackers. No obstante, lo mejor es conocer bien nuestro propio sistema para poder adaptar estas medidas a nuestro caso concreto, para lo cual se debe tener en cuenta lo siguiente:

Sólo la Información Necesaria: No almacene información sensible en su ordenador si ésta no necesita ser consultada desde el exterior. ¿Por qué colocar un premio extra para los hackers?.

Instalación de demonios: Cuando instale cualquier software que incluya algún demonio, asegúrese de que se trata de la versión más reciente y actualizada, que debería ser la más segura. Desconfíe de las versiones beta, a no ser que sepa muy bien lo que hace. Configure sus servidores de la forma más conservadora posible. No habilite usuarios genéricos sin antes asegurarse de que no poseen excesivos privilegios. Si tiene alguna duda sobre alguna funcionalidad del servidor en cuestión, deshabilítela. Consulte periódicamente las páginas de los fabricantes de software y aquellas especializadas en alertar sobre fallos de seguridad, ellas le informarán de los agujeros más recientes y de cómo eliminarlos. Muchos hackers también las

consultan, pero con otros propósitos. Ejecute periódicamente alguna utilidad que recorra los puertos de su sistema para saber en cuáles hay demonios. Esto le permitirá detectar programas del tipo Back Office.

Vigile su software Criptográfico: Emplee siempre que pueda SSL o TLS en sus comunicaciones, y asegúrese de que todos los programas de cifrado que usa funcionan con claves de al menos 128 bits.

Contra los ataques por Diccionario: Muchos sistemas operativos impiden que un administrador abra una consola remota, por lo que aunque alguien averigüé su contraseña, no podrá emplearla a no ser que disponga de acceso físico a la computadora. Si su ordenador está conectado a Internet, use esta característica, a no ser que necesite poder abrir consolas remotas como administrador.

Los Archivos de Registro: Serán nuestra mejor defensa contra los hackers. Hay que consultarlos frecuentemente para detectar entradas sospechosas, y nunca bajar la guardia. Tampoco viene mal efectuar copias de seguridad en medios externos al ordenador, como pueden ser diskettes o cintas magnéticas. De esta forma el vaquero no podrá borrar totalmente. Asegúrese de que el fichero de contraseñas está protegido frente a accesos externos. Afortunadamente, casi todos los sistemas operativos modernos incorporan esta característica por defecto. Cambie periódicamente las contraseñas, y sobre todo, use buenas contraseñas.

4.3. Redes VPN

Una RED se extiende sobre un área geográfica amplia, a veces un país o un continente; contiene una colección de máquinas dedicadas a ejecutar programas de usuario (aplicaciones). En los últimos años las redes se han convertido en un factor

crítico para cualquier organización. Cada vez en mayor medida, las redes transmiten información vital, por tanto dichas redes cumplen con atributos tales como seguridad, fiabilidad, alcance geográfico y efectividad en costos.

Se ha demostrado en la actualidad que las redes reducen en tiempo y dinero los gastos de las empresas, eso ha significado una gran ventaja para las organizaciones sobre todo las que cuentas con oficinas remotas a varios kilómetros de distancia, pero también es cierto que estas redes remotas han despertado la curiosidad de algunas personas que se dedican a atacar los servidores y las redes para obtener información confidencial. Por tal motivo la seguridad de las redes es de suma importancia, es por eso que se habla tanto de los famosos Firewalls y las VPN.

4.3.1. ¿Por qué una VPN?

Cuando deseo enlazar mis oficinas centrales con alguna sucursal u oficina remota tengo tres opciones:

Modem: Las desventajas es el costo de la llamada, ya que el costo de esta llamada sería por minuto conectado, además sería una llamada de larga distancia lo que incrementaría su costo, a parte no contaría con la calidad y velocidad adecuadas. Los Modems tienen la desventaja de no brindar velocidades altas de conexión.

Línea Privada: Tendría que tender mi cable ya sea de cobre o fibra óptica de un punto a otro y a los puntos en donde me voy a conectar desde la sede principal, en esta opción el costo es muy elevado porque si por ejemplo necesito enlazar mi oficina central con una sucursal que se encuentra a 200 Kilómetros de distancia el costo sería por la renta mensual por Kilómetro. Sin importar el uso que se de al PVC independientemente del tráfico.

VPN: Los costos son bajos porque solo realizo llamadas locales, además de tener la posibilidad de que mis datos viajen ***encriptados*** y seguros, con una buena calidad y velocidad.

4.3.2. ¿Qué es una VPN?

Es una red privada que se extiende, mediante un proceso de encapsulación y en su caso de encriptación, de los paquetes de datos a distintos puntos remotos mediante el uso de unas infraestructuras públicas de transporte.

Los paquetes de datos de la red privada viajan por medio de un "túnel" definido en la red pública (ver Figura 4.2).

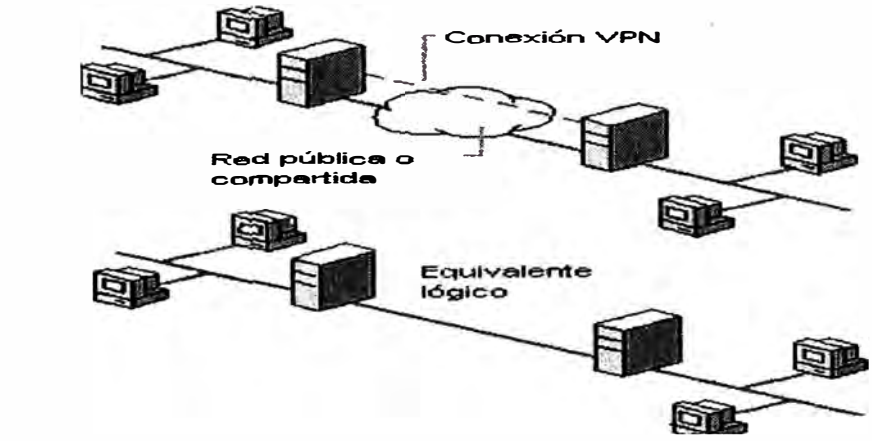

Figura 4.2. Conexión de una VPN

Figura 4.3. Datos a través de una VPN

En la figura anterior (ver Figura 4.3) se muestra como viajan los datos a través de una VPN ya que el servidor dedicado es del cual parten los datos, llegando a un Firewall que hace la función de una pared para engañar a los intrusos a la red, después los datos llegan a nube de Internet donde se genera un túnel dedicado únicamente para nuestros datos para que estos con una velocidad garantizada, con un ancho de banda también garantizado y lleguen a su vez al Firewall remoto y terminen en el servidor remoto.

Las VPN pueden enlazar mis oficinas corporativas con los socios, con usuarios móviles, con oficinas remotas mediante los protocolos como Internet, IP, Ipsec, Frame Relay, ATM como lo muestra la Figura 4.4 siguiente.

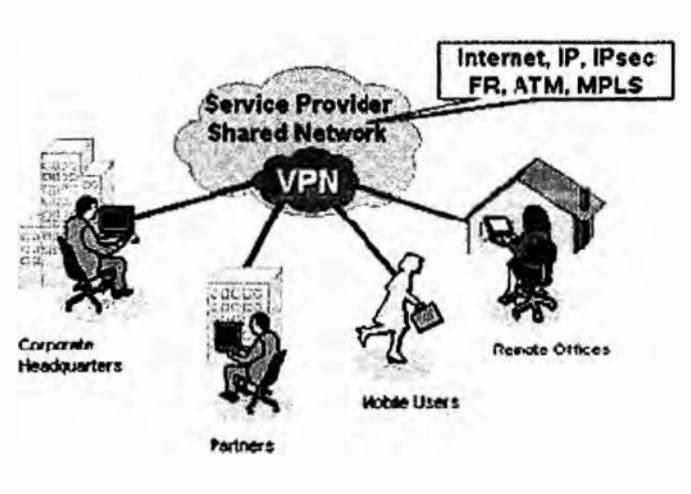

Figura 4.4. Uso de tecnologías para una VPN

4.3.3. Topología Túnel

Las redes privadas virtuales crean un túnel o conducto de un sitio a otro para transferir datos, a esto se le conoce como encapsulación además los paquetes van encriptados de forma que los datos son ilegibles para los extraños, usando para esto diferentes protocolos de seguridad y encriptación.

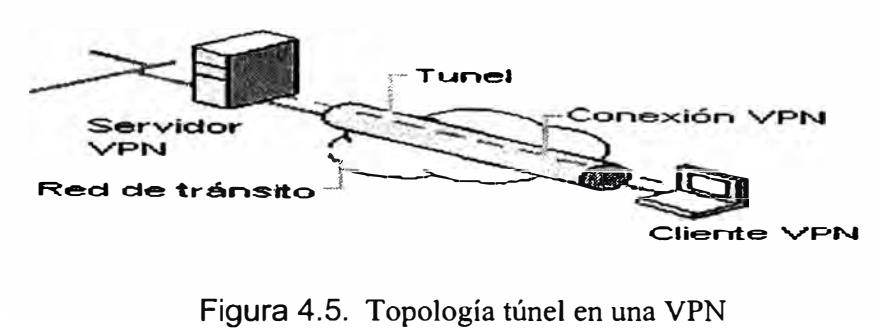

Figura 4.5. Topología túnel en una VPN

El servidor busca mediante un ruteador la dirección IP del cliente VPN y en la red de transito se envían los datos sin problemas (ver Figura 4.5).

4.3.4. Requerimientos básicos de una VPN

Por lo general cuando se desea implantar una VPN hay que asegurarse que esta proporcione:

Identificación de usuario: La VPN debe ser capaz de verificar la identidad de los usuarios y restringir el acceso a la VPN a aquellos usuarios que no estén autorizados. Así mismo, debe proporcionar registros estadísticos que muestren quien acceso, que información y cuando.

Administración de direcciones: La VPN debe establecer una dirección del cliente en la red privada y debe cerciorarse que las direcciones privadas se conserven así.

Codificación de datos: Los datos que se van a transmitir a traves de la red pública deben ser previamente encriptados para que no puedan ser leídos por clientes no autorizados de la red.

Administración de claves: La VPN debe generar y renovar las claves de codificación para el cliente y el servidor.

Soporte a protocolos múltiples: La VPN debe ser capaz de manejar los protocolos comunes que se utilizan en la red pública. Estos incluyen el protocolo de Internet (IP), el intercambio de paquete de Internet (IPX) entre otros.

4.3.5. Herramientas de una VPN

Las herramientas necesarias para implementar una VPN son las siguientes:

- VPN Gateway.
- Software VPN Cliente.
- Firewall.
- Router.

Dispositivos con un software y hardware especial para proveer de capacidad a la VPN. Esta sobre una plataforma PC ó Workstation, el software desempeña todas las funciones de la VPN.

4.3.6. Ventajas de una VPN

Dentro de las ventajas más significativas podremos mencionar:

- La integridad, confidencialidad y seguridad de los datos.
- Reducción de costos.
- Sencilla de usar.
- Sencilla instalación del cliente en cualquier PC Windows.
- Control de Acceso basado en políticas de la organización
- Herramientas de diagnostico remoto.
- Los algoritmos de compresión optimizan el tráfico del cliente.
- Evita el alto costo de las actualizaciones y mantenimiento a las PC´s remotas.
- Múltiples programas de aplicación para clientes.
- Es posible trabajar diferentes tipos de marcas de equipos.

4.3.7. VPN Full Mesh por Internet

Si tenemos varias sucursales a través del mundo y estas conectadas al Internet. Entonces podemos implementar una Red VPN entre nuestras sucursales haciendo uso del Internet. Esto implica una re-configuración de los equipos (Firewall o Router) para formar que las conexiones VPN y formar una red que sea una malla completa (Full-Mesh), es decir que todas las sedes pueden interconectarse entre sí de manera directa vía las VPN. Obviamente que el costo de estas conexiones no se asumen dado que los datos son transportados a través del Internet. De igual manera podemos hacer uso de aplicativos VPN Client para brindar acceso seguro y encriptado a nuestra red interna a los usuarios desde cualquier parte del Internet. Esto definitivamente brinda un alto grado de seguridad y confiabilidad a nuestra red.

CONCLUSIONES

1. Después de todo lo estudiado parece una locura conectarse a una red externa, y ciertamente lo es si no se toman las precauciones adecuadas. La cantidad de posibles riesgos es enorme, y con toda seguridad en el futuro aparecerán nuevos peligros, pero no olvidemos que ante todo debemos ser racionales. Si bien puede ocurrir que un equipo de hackers trate de entrar en nuestro sistema, esta posibilidad suele ser remota en la mayoríaia de los casos, debido precisamente al escaso interés que va a despertar en ellos penetrar en una red pequeña. No olvidemos que el coste de la protección en ningún caso puede superar el valor de la propia información que se desea proteger. Por lo demás, parece claro que las ventajas que nos proporcionaría estar en la Red son claramente mayores que los inconvenientes, pero nunca se debe bajar la guardia.

2. En general, conviene estar preparado para el peor de los casos probables, que suele ser la pérdida de la información, casi siempre debida a fallos físicos o a la presencia de virus. También es claro que se debe elegir la adopción de protocolos seguros, además de llevar un registro de todas las operaciones que tienen lugar dentro del sistema, registro que debería ser controlado periódicamente para detectar posibles anomalías. Otra práctica bastante recomendable consiste en mantenerse al

día sobre los fallos de seguridad detectados en los programas y sistemas operativos que empleemos, asíı como de los sucesivos parches que las empresas de software suelen distribuir periódicamente, con objeto de eliminar los agujeros de seguridad.

4. Cuando hacemos un viaje solemos tomar ciertas precauciones, que reducen el riesgo de tener un accidente, o al menos el posible daño en caso de sufrirlo. Nadie que esté lo suficientemente equilibrado mentalmente y que conozca los medios de transporte emprende un viaje pensando que va a sufrir un accidente. Estos razonamientos son totalmente válidos cuando nos enfrentamos a los hackers. No hay que alarmarse, pero tampoco hemos de bajar la guardia.

ANEXO A

ÍNDICE DE ILUSTRACIONES

ANEXO B

ÍNDICE DE TABLAS

BIBLIOGRAFÍA

[1] Mendivil, Ignacio. El ABC de los documentos electrónicos seguros. Oct, 1999.

[2] J.J. Ángel Á. Generación de Números Pseudoaleatorios usados en Sistemas Criptográficos http://www.seguridata.com. Crypto Notas Vol. 1, No. 1, 1998.

[3] FIPS 140, "Security requirements for cryptographic modules", Federal Information Processing Standards Publication 140 - I, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Virginia, 1994.

[4] Lucena López, José. Criptografía y Seguridad en computadores. Universidad de Jaen, España. May, 2003.

[5] C.K. Koc, High Speed RSA Implementation, RSA Laboratories, RSA Data Security, Inc., 100 Marine Parkway City, CA 94065, 1995.

[6] Seberry, J., Pieprzyk, J. Cryptography. An Introduction to Computer Security. Prentice Hall.

[7] Arora, Rakesh. Voice over IP: Protocols and Standards. http://www.cis.ohio-state.edu/~jain/cis788-99/voip_protocols/index.html. Ohio State University.

[8] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Florida, 1996.

[9] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, New York, 2nd edition, 1996.

[10] Página Web de Kriptópolis. http://www.kriptopolis.com

[11] RFC /1750: Randomness Recommendations for Security.