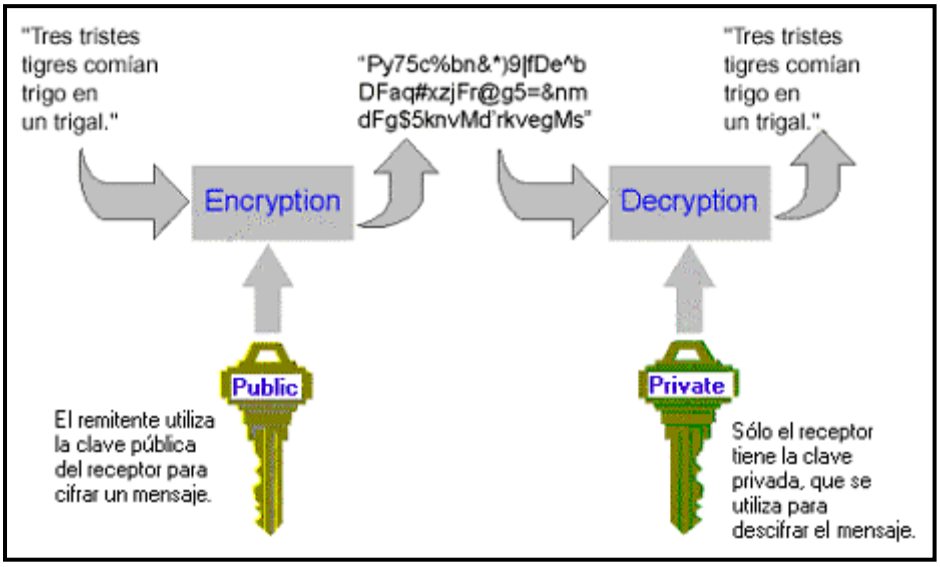
C A P Í T U L O III



CRIPTOGRAFÍA Y SEGURIDAD

3.1 Introducción a la Criptografía

3.2 Sistemas y mecanismos de Encriptación

3.3 Algoritmos y Funciones Criptográficas

3.4 Algoritmos de llaves simétricas

3.5 Algoritmos de llaves públicas

3.6 Criptografía y el web

3.7 Estándares criptográficos y electrónicos aplicables

3.8 Comprobación de Integridad de Mensajes

3.9 Escuchas Electrónicas

3.10 Crackers y Hackers

3.1 Introducción a la Criptografía

La Criptografía es el: "Arte de escribir con clave secreta o de un modo enigmático". Existen dos documentos fundamentales que sientan las bases de la Teoría de la Información, y que junto con otro artículo posterior del mismo autor sirvió de base para la Criptografía moderna, uno escrito por Claude Shannon en 1948 "A Mathematical Theory of Communication". El otro, publicado por Whitfield Die y Martin Hellman en 1976, "New directions in Cryptography", que introdujo el concepto de Criptografía de Llave Pública, dando el punto de partida al estudio de esta ciencia. La palabra Criptografía sólo se refiere al uso de códigos para cifrar documentos, por lo que no engloba a las técnicas que se usan para romper dichos códigos (Criptoanálisis). De ahí nace el término Criptología, que se encarga del estudio tanto de la Criptografía como del Criptoanálisis.

La criptografía es un conjunto de técnicas utilizadas para conservar segura la información, con ella es posible transformar palabras escritas y otros tipos de mensajes de forma que sean incomprensibles para receptores no autorizados. Un receptor autorizado puede después regresar las palabras o mensajes a un mensaje perfectamente comprensible. Esta técnica utiliza algoritmos y funciones matemáticas que permiten transformar la información en un conjunto de símbolos ilegibles y también permiten el proceso contrario, transformar el conjunto de símbolos ilegibles en textos legibles. El origen de la criptografía es muy antiguo, los generales griegos y romanos por ejemplo, la utilizaban para enviar mensajes en clave a los comandantes que estaban en el campo de batalla, estos sistemas primitivos se basaban en ciertas técnicas:

- ✓ ***La Sustitución.-*** cuyo principio es reemplazar cada letra del mensaje que se desea encriptar con otra. Por ejemplo la letra "a" por la "d", la "b" por la "e", etc.
- ✓ ***La Transposición.-*** que se basa en la revoltura[39] de los caracteres del mensaje, que implica escribir un mensaje dentro de una tabla, renglón por renglón y luego leerlo columna por columna.
- ✓ ***Doble Transposición.-*** similar a la transposición con la diferencia que se necesita repetir la revoltura otra vez.

Durante años la criptografía se ha desarrollado de forma paralela a la tecnología militar que fue la que con mayor énfasis hizo uso de la criptografía por lo que fue considerada como una tecnología militar, sin embargo casi todos los sistemas criptográficos han sido inventados por civiles y usados en distintos fines tales como: secretos militares, religiosos, diplomáticos, científicos, etc. En la actualidad la criptografía es la primera herramienta usada dentro de los negocios y el comercio electrónico a través del Internet logrando un gran desarrollo y avance en los negocios a través de medios electrónicos. La criptografía es una tecnología de uso dual, que tiene aplicaciones tanto militares como civiles, para todos los usuarios la criptografía es una forma de comprar seguridad y reducir el riesgo en un mundo de incertidumbres. [LIB 05]

Criptografía Moderna

Cuando se desea transmitir información entre computadores, especialmente a través de redes públicas, se plantea la necesidad de que la transmisión sea segura. La seguridad de la información transmitida tiene dos aspectos:

[39] Proceso de ocultar el mensaje que se quiere enviar, como si se lo metiese dentro de una protección.

- ✓ Que sea el destinatario del mensaje el único que pueda leerlo.
- ✓ Que nadie pueda suplantar la identidad de emisor del mensaje o modificar el contenido de los mensajes que envía.

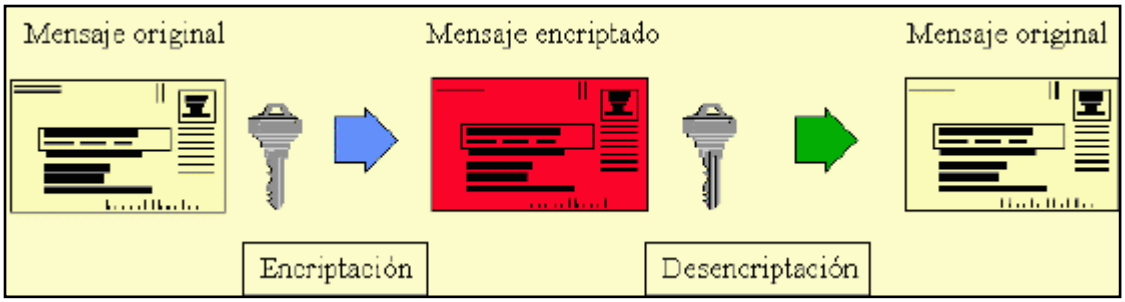


Figura 3.1 Principio de la Criptografía

La Criptografía se basa en dos procesos complementarios

- ✓ **La Encriptación.-** proceso por el cual un mensaje legible, es transformado en un mensaje ilegible (encriptación o cifrado), salvo para su destinatario o cifrado mediante una transformación matemática y una clave secreta denominada clave de encriptación
- ✓ **Desencriptación.-** proceso inverso a la encriptación; a partir de un texto cifrado y una clave secreta se reconstruye el texto original. Con unos sistemas, la clave de encriptado es la misma que para desencriptar; con otros sistemas, las claves son distintas.

Todos los sistemas criptográficos, sin importar cuan complejos tienen las siguientes partes básicas.

- ✓ **Plaintex (texto en claro).-** es el mensaje antes de que se haga cualquier cosa, Este es entendible por los humanos en un formato que cualquier software adecuado pueda usar.
- ✓ **Texto Cifrado.-** es el mensaje en plaintext después de que ha sido modificado para hacerlo ilegible. El proceso de convertir texto en claro a texto cifrado se dice encriptar y la operación inversa es desencriptar.

- ✓ **Algoritmo de encripción.-** es la operación usada para convertir texto en claro a texto cifrado y viceversa.
- ✓ **Llave.-** es una llave secreta usada para encriptar o desencriptar el mensaje. Cada llave transforma el mismo texto en claro en diferente texto cifrado. Si el sistema criptográfico funciona bien, solamente las personas que conocen la llave correcta pueden descifrar el texto cifrado. [WWW 025]

3.2 Sistemas y mecanismos de Encriptación

La criptografía incluye técnicas como de esconder información almacenada o en tránsito. Hoy en día la criptografía se asocia más a convertir texto sencillo a texto cifrado y viceversa. La Criptografía se ocupa de dar solución a los problemas de identificación, autenticación y privacidad de la información en los sistemas informáticos. En la medida en que han ido creciendo las redes, el intercambio seguro de las claves secretas se ha vuelto costoso y problemático.

Algoritmo.- *en general es la serie de reglas que no pueden ser de doble sentido y deben tener una meta clara. Los algoritmos pueden ser expresados en cualquier lenguaje, y en cualquier lenguaje de programación de computadoras. Los algoritmos criptográficos son la base para construir aplicaciones y protocolos de encripción.*

Existen dos tipos generales de algoritmos basados en claves que son

1. **Algoritmos de Encripción Simétricos**

   Cuando la clave que va a encriptar el mensaje puede ser calculada desde la clave para desenciptar y viceversa se le conoce como algoritmo simétrico. En muchos de los algoritmos asimétricos, la clave de encripción y para desencriptar es la misma. Estos algoritmos requieren que el emisor y el receptor tengan la misma

clave antes de comunicarse. La seguridad de un algoritmo simétrico realmente recae en la clave. El divulgar la clave significa que cualquiera puede encriptar o desencriptar la información. La clave tiene que mantenerse en secreto tanto tiempo como la comunicación se quiere mantener en secreto.

2. **Algoritmos de Encripción Asimétricos**

   Son diseñados de tal manera que una clave se usa para encriptar y una diferente para desencriptar. Esto ocasiona que teniendo la clave para desencriptar, no se puede calcular la clave de encripción. Estos algoritmos son llamados de "clave pública" porque la clave para encripción se puede publicar. Una persona cualquiera, puede usar la clave para encriptar el mensaje, pero sólo una persona puede desencriptar el mensaje. En estos sistemas, la clave de encripción es llamada clave pública y la clave para desencriptar se llama clave privada. [LIB 01]

Métodos de Encripción.- Cada uno de los algoritmos es identificado por un nombre, un propósito, un rango de clave y por la fecha de creación. Todos los algoritmos tienen uno o más propósitos:

- ✓ **Encripción.-** los algoritmos se usan simplemente para encriptar comunicación. Tanto el emisor como el receptor encriptan y desencriptan el mensaje usando el mismo algoritmo.
- ✓ **Firmas Digitales.-** son algoritmos de clave pública con información secreta para firmar documentos e información pública para verificar las firmas. Al proceso de firmado se conoce como encriptar con una clave privada y la verificación se conoce como desencriptar con una clave pública.
- ✓ **Hashing y Digest.-** funciones matemáticas que toman una cadena de longitud variable y la convierten a una cadena de

longitud fija. Es una manera de obtener una huella digital de los datos. El algoritmo de hashing genera un valor para el mensaje. El Digest es la representación del texto en forma de una cadena de dígitos, creado con una fórmula de hashing de una sola dirección. El encriptar un digest de un mensaje con una clave privada, genera una firma digital. Usando criptografía de clave pública, el emisor del mensaje cifrará el mensaje aplicando la clave pública del destinatario. Será por tanto el destinatario, el único que podrá descifrar el mensaje aplicando su clave privada.

Infraestructura para llaves públicas

El principal problema que tiene la firma digital es que no se ha convertido en un estándar global debido a razones políticas (en muchos países no tiene el mismo peso legal que una firma normal, y en muchos de ellos todavía no se ha masificado el software necesario para su implementación y utilización)

Una firma digital es un conjunto de datos asociados a un mensaje que permite asegurar la identidad del firmante y la integridad del mensaje. La firma digital no implica que el mensaje esté cifrado, esto es, un mensaje firmado será legible en función de que esté o no cifrado. El firmante generará mediante una función, un 'resumen' o huella digital del mensaje. Este resumen o huella digital la cifrará con su clave privada y el resultado es lo que se denomina firma digital, que enviará adjunta al mensaje original. Cualquier receptor del mensaje podrá comprobar que el mensaje no fue modificado desde su creación porque podrá generar el mismo resumen o misma huella digital aplicando la misma función al mensaje. Además podrá comprobar su autoría, descifrando la firma digital con la clave pública del firmante, lo que dará como resultado de nuevo el

resumen o huella digital del mensaje. La criptografía de clave pública o asimétrica esta basada en el uso de un par de claves que lo que se puede cifrar con una de ellas, solo se puede descifrarlo con la otra y sólo con ella. Una de las claves solo está en poder del propietario, que debe conservarla de forma segura, y se denomina clave privada. La otra clave es publicada para que la conozcan todos aquellos que quieran comunicarse de modo seguro con el propietario mencionado, a esta última se la denomina clave pública.

3.3 Algoritmos y Funciones Criptográficas

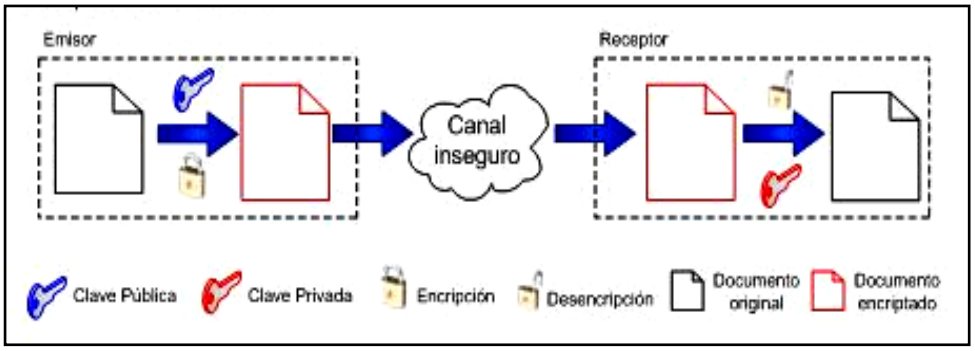


Figura 3.2 Principio de la Infraestructura de Clave Pública

Los algoritmos de encriptación actuales se agrupan en dos grandes clases según el proceso de encriptación y descifrado, si se usa la misma clave se denominan algoritmos de claves simétricas; si se utilizan claves distintas se denominan algoritmos de claves asimétricas o algoritmos asimétricos.

a) **Algoritmos de llaves simétricas:** En este tipo de algoritmos se utiliza la misma llave para encriptar y desencriptar el mensaje. Estos algoritmos se conocen también como algoritmos de llave privada.

b) **Algoritmos de llave pública:** En este tipo de algoritmos se utiliza una llave para encriptar el mensaje y otra llave para desencriptar el mensaje, la llave de encriptación por lo general se

conoce como ***llave pública***[40]***,*** ya que puede ser divulgada al público sin poner en peligro la confidencialidad del mensaje ni la llave de desenciptación. La llave de desenciptación es conocida como ***llave privada***[41] ***o secreta.***

La seguridad de la criptografía depende de los siguientes parámetros

- ✓ Que la clave sea realmente secreta.
- ✓ Longitud de la clave
- ✓ Que el algoritmo no sea invertible; es decir, que si se conoce cómo funciona, no se pueda dar la vuelta al proceso sin la llave
- ✓ Que el algoritmo no permita descifrar todo el texto si se conoce el contenido de una parte

Propiedades de los algoritmos de clave pública

Según Diffie y Hellman, todo algoritmo de clave pública debe cumplir las siguientes propiedades de complejidad computacional:

- ✓ Cualquier usuario puede calcular sus propias claves pública y privada en tiempo polinomial.
- ✓ El emisor puede cifrar su mensaje con la clave pública del receptor en tiempo polinomial.
- ✓ El receptor puede descifrar el criptograma con la clave privada en tiempo polinomial.
- ✓ La persona criptoanalista que intente averiguar la clave privada mediante la pública se encontrará con un problema intratable.
- ✓ La persona criptoanalista que intente descifrar un criptograma teniendo la clave pública se encontrará con un problema intratable.

c) **Algoritmos de Hashing:** Son una parte esencial en casi todas las soluciones criptográficas del mundo real. El propósito de estos algoritmos es reducir un texto de cualquier longitud a otro de una longitud fija llamado resumen, digesto o hash cumpliéndose que:

- ✓ Dos textos distintos no pueden tener un mismo hash. (libre de colisiones)

[40] Llave que puede ser conocida por todo el mundo
[41] Que solo la puede conocer el propietario de la Firma.

- ✓ No se puede recuperar el texto original desde un hash. (irreversible)

En general estos algoritmos se utilizan para:

- ✓ **Comprobar la integridad de un mensaje.-** con solo cambiar un bit del texto original, el hash cambia radicalmente de forma impredecible, por lo tanto si se transmite un mensaje junto con su hash, el receptor puede recalcularlo y compararlo con el recibido, si estos difieren quiere decir que el mensaje fue modificado en el camino. Para este tipo de usos se recomienda HMACSHA1 o MACTripleDES ya que requieren de una clave secreta (solo conocida por el emisor y el receptor) para le generación del hash por lo que si alguien quiere cambiar el mensaje en el camino (tampering) no podrá pasar inadvertido tan solo reemplazando el hash original con uno calculado a partir del mensaje modificado. [WWW 026]
- ✓ **Proteger datos como contraseñas en una base de datos.-** es muy recomendable que en lugar de guardar las contraseñas de los usuarios en una base de datos o archivo de configuración, se guarde el hash de las mismas, evitando que si alguien no autorizado logra acceder a la BD pueda hacerse de los accesos al sistema. Por este motivo ASP.NET provee una función específica: FormsAuthentication.HashPasswordForStoringInConfigFile.

3.4 Algoritmos de llaves simétricas

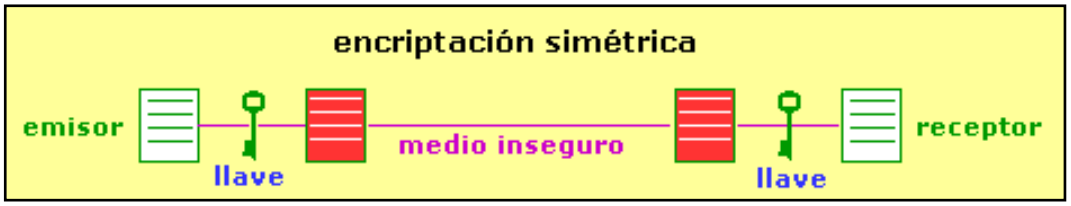


Figura 3.3 Esquema de Llaves Simétricas

La encriptación y descifrado con claves simétricas es entre diez y cien veces más rápido que un algoritmo de claves asimétricas (claves distintas). Su esquema de trabajo con claves simétricas se muestra en la Figura 3.3. En esta imagen, un texto "plain text" es transformado según una clave (Ksecret), trasferido, y desencriptado con la misma palabra clave. Esta clave debe haberse suministrado por un medio seguro al receptor, de manera que no haya sido "capturada" o robada por algún intruso.

a) **Algoritmo DES**

Adoptado como estándar ANSI en 1977. El Data Encryption Standard (DES) o el Algoritmo de Encripción de Datos (DEA) fue un estándar por cerca de 20 años. Este algoritmo se ha desempeñado muy bien y ofrece seguridad. DES es un bloque cifrado, encriptando los datos en bloques de 64 bits, el mismo algoritmo se usa para encriptar y desencriptar la información.

DES es un método de cifrado altamente resistente frente a ataques criptoanalíticos diferenciales, ya que transforma segmentos de mensaje de 64 bits en otros equivalentes de texto cifrado, empleando una clave de 56 bits. Sin embargo, es vulnerable frente a los potentes métodos de decodificación posibles para los grandes ordenadores y no ha sido considerado apropiado para las aplicaciones recientemente realizadas. En la actualidad ofrece protección contra el pirata informático habitual, pero

no contra un esfuerzo masivo por parte de un usuario con grandes recursos.

DES fue el primer sistema de cifrado en bloque que se extendió al sector público, por lo que es usado a menudo para describir nuevas técnicas de criptoanálisis. Incluso hoy, no existe ninguna técnica de criptoanálisis que pueda vulnerar completamente DES de un modo estructural; la única debilidad de DES es el pequeño tamaño de la clave y quizá el pequeño tamaño del bloque. DES maneja lo que se conoce como **Claves débiles**[42]**;** La misma sub-clave es generada en cada iteración. DES tiene 4 claves débiles. Y las **Claves semi-débiles**[43]: Sólo dos sub-claves se generan en iteraciones alternadas. DES maneja doce claves de este tipo en seis pares. [WWW 028]

Funcionamiento.- DES cifra bloques de datos de 64 bits usando una clave de 56 bits. Normalmente se tiene una cantidad de información arbitraria para cifrar y se necesita una forma de especificar cómo se realiza ese cifrado. La manera en que se usa un cifrador en bloque se denomina **modo de uso** y para DES estándar ANSI ha definido cuatro, dos en bloque y dos en flujo:

En bloque

- ✓ **Electronic Codebook Book (ECB)**: El mensaje se divide en bloques independientes de 64 bits y el cifrado se efectúa bloque a bloque.

  C(i) = DES(K1) (P(i))

[42] Las Misma Clave es utilizada durante todo el proceso de cifrado, por lo que es fácil saberla.
[43] Se utilizan dos subclaves durante todo el proceso, es más difícil conocer la clave.

- ✓ **Cipher Block Chaining (CBC)**: De nuevo el mensaje se divide en bloques de 64 bits, pero estos se unen en el cifrado mediante un vector de inicialización IV.

  C(i) = DES(K1) (P(i)(+)C(i-1)), con C(-1)=IV

En flujo

- ✓ **Cipher FeedBack (CFB)**: Los bits del mensaje son añadidos a la salida del DES, y el resultado se lleva al siguiente bloque. Requiere también de un vector de inicialización.

  C(i) = P(i)(+) DES(K1) (C_(i-1)), con C_(-1)=IV

- ✓ **Output FeedBack (OFB)**: Es igual que el anterior pero sin realimentación.

  C(i) = P(i)(+) O(i) O(i) = DES(K1)(O(i-1)), con O(-1)=IV

  Cada modo de cifrado presenta sus ventajas y sus desventajas.

Características:

- ✓ Propósito: Encripción
- ✓ Rango de clave: 56 bits
- ✓ Fecha de Creación: 1976

b) **Algoritmo Triple-DES**

Algoritmo similar a DES, con la diferencia que utiliza tres claves distintas. Es el más utilizado en transacciones en instituciones financieras. Basado en tres iteraciones sucesivas del algoritmo DES, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con DES simple. Este hecho se basa en que DES tiene la característica matemática de no ser un grupo, lo que implica que si se encripta el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave.

Funcionamiento.- Se toma una clave de 128 bits y se divide en 2 diferentes de 64 bits, aplicándose el siguiente proceso al documento en claro:

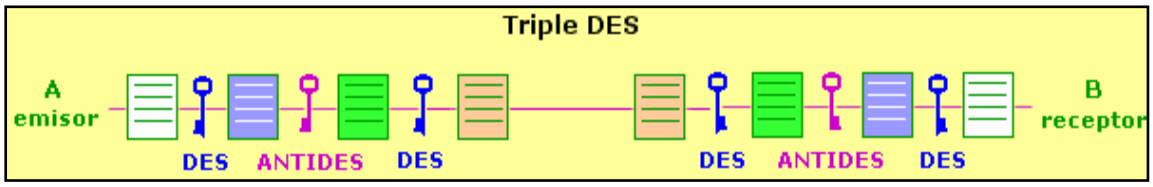


Figura 3.4 Esquema del Algoritmo Triple DES

Se le aplica al documento a cifrar un primer cifrado mediante la primera clave, C1. Al resultado (denominado ANTIDES) se le aplica un segundo cifrado con la segunda clave, C2. Y al resultado se le vuelve a aplicar un tercer cifrado con la primera clave, C1. Si la clave de 128 bits está formada por dos claves iguales de 64 bits (C1=C2), entonces el sistema se comporta como un DES simple, actualmente TDES usa 3 claves diferentes, lo que hace el sistema mucho más robusto, al conseguirse longitudes de clave de 192 bits (de los cuales son efectivos 168), mientras que el uso de DES simple no está aconsejado. [WWW 027]

c) **Algoritmo RC2**

Sistema desarrollado por Ronald Rivest. Que realiza el cifrado en bloques, adoptado inicialmente por la agencia RSA; admite claves con longitudes entre 1 y 2048 bits. La versión de exportación limita su uso a claves de 40 bits.

d) **Algoritmo RC4**

Sistema de cifrado de flujo, también adoptado por por la agencia RSA; admite claves con longitudes entre 1 y 2048 bits. La versión de exportación limita su uso a claves de 40 bits. Desarrollado por Ronald Rivest en 1994.

e) **Algoritmo RC5**

Sistema de cifrado en bloques adoptado inicialmente por la agencia RSA; admite claves con longitudes entre 1 y 2048 bits. Permite que el usuario varíe el tamaño del bloque que se encripta en cada paso. Desarrollado por Ronald Rivest en 1994.

f) **Algoritmo DESX**

Este algoritmo es una variación del DES; introduce un proceso de encriptado en dos fases que hace prácticamente imposible encontrar la clave.

3.5 Algoritmos de llaves públicas

Los algoritmos asimétricos o de Clave Pública, son lentos pero tienen la ventaja de que una de las claves puede ser conocida por cualquiera; el mensaje encriptado por esa clave "pública" sólo puede ser descifrado por la otra clave, privada, conocida sólo por el destinatario.

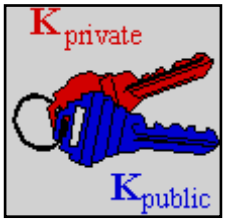


Figura 3.5 Estructura de la PKI (Una llave pública y una Privada)

Si A desea enviar un mensaje a B, A encripta el mensaje con la clave pública (KB,public) de B; al recibir el mensaje, B lo descifra con su clave privada (KB,private). Por el contrario, si B desea enviar un mensaje a A, B encripta el mensaje con la clave pública (KA,public) de A; al recibir el mensaje, A lo descifra con su clave privada (KA,private)

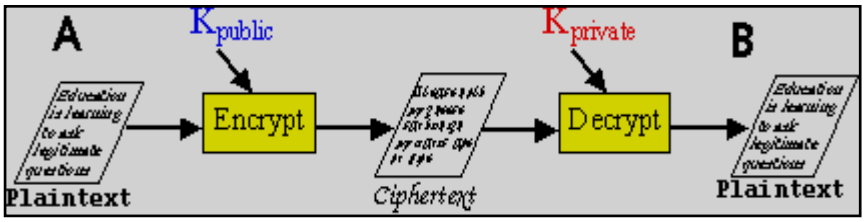


Figura 3.6 Proceso de Encriptación y Desencriptación

Existen algoritmos asimétricos para bloques de texto, que trabajan sobre un número de bytes, en tamaños definidos; y algoritmos asimétricos de flujo de datos, que encriptan byte a byte toda la información. El sistema de firma digital (digital signature system) fue desarrollado por la Agencia de Seguridad Nacional de los EE. UU. (NSA) y puede utilizar claves entre 512 y 1024 bits de longitud. La seguridad de estos sistemas depende de la longitud de la clave.

Paso 1: A genera la clave simétrica (Ks) que deben utilizar A y B en sus transmisiones en esa sesión. Las razones para utilizar una clave simétrica son la velocidad y que, para mayor seguridad, se genera para el caso. Para enviar esta clave de manera segura a B, A utiliza la clave asimétrica pública de B (Kb, public); cuando B recibe la clave Ks encriptada con su clave pública, la descifra con la clave asimétrica privada (Kb, private) que sólo posee B.

Paso 2: Una vez que A y B poseen la misma clave simétrica (Ks) la trasmisión se realiza encriptando toda la información con esa clave en ambos sentidos.

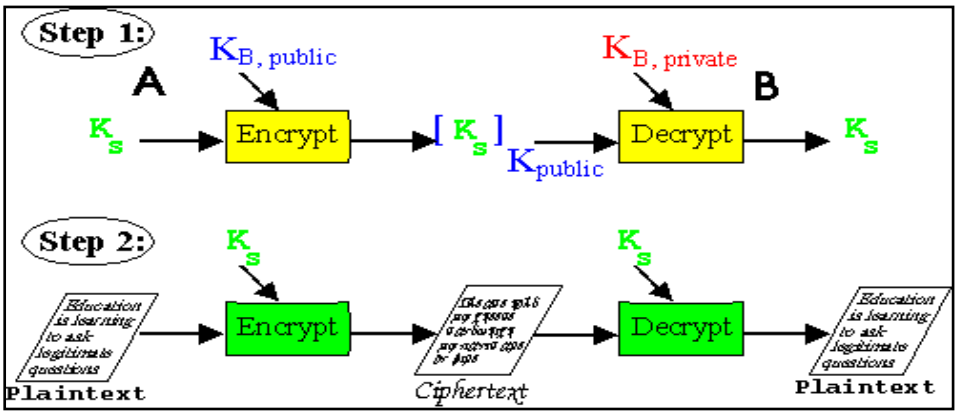


Figura 3.7 Proceso de Transmisión de un mensaje Encriptado

Digest o funciones resumen para la criptografía

Un digest es una función matemática que produce una secuencia de caracteres, normalmente entre 128 y 256 bits de longitud, a partir de un archivo de información inicial de cualquier longitud.

Propiedades

- ✓ Al resumen de un mensaje, se la llama generalmente huella digital del mensaje.
- ✓ Cada bit de salida del digest es influenciado por cada bit de entrada
- ✓ Para cualquier bit que se cambie a la entrada, cada bit de salida tiene al menos un 50% de probabilidades de cambiar
- ✓ Dado un texto de entrada, debe ser informáticamente viable encontrar otro texto de entrada que del mismo producto en el digest.
- ✓ Dos mensajes iguales producen huellas digitales iguales
- ✓ Dos mensajes parecidos producen huellas digitales completamente diferentes.
- ✓ Dos huellas digitales idénticas pueden ser el resultado de dos mensajes iguales o de dos mensajes completamente diferentes.
- ✓ Una función hash es irreversible, no se puede deshacer, por tanto su comprobación se realizará aplicando de nuevo la misma función hash al mensaje.

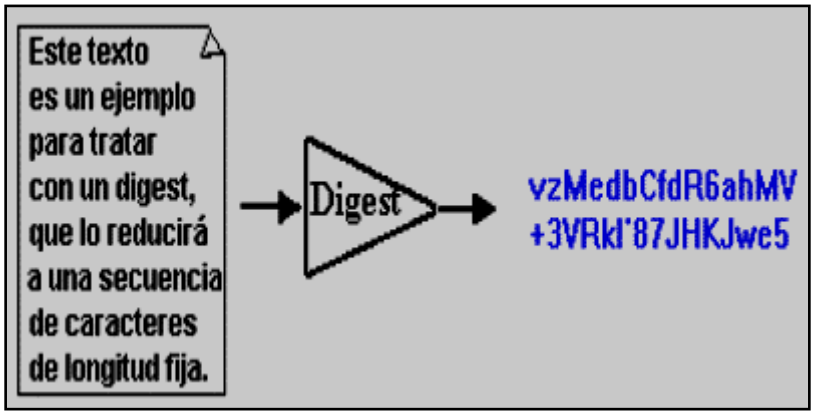


Figura 3.8 Proceso de transformación de un Digest a un Texto Normal

Funciones digest más utilizadas

- ✓ HMAC.- Hassed Message Authentication Code, usa una clave secreta

- ✓ MD2.- Message Digest #2, desarrollado por Ronald Rivest, produce un digest de 128 bits, requiere mucho tiempo para su cálculo.
- ✓ MD4.- Desarrollado por Ronald Rives como alternativa al MD2. Ha resultado ser un tanto inseguro.
- ✓ MD5.- Más seguro que el MD4, genera un digest de 128 bits
- ✓ SHA.- Desarrollado por la NSA, produce un digest de 160 bits

Siempre es posible utilizar un digest dentro de un mensaje y encriptar el digest con una clave; de esta forma se puede firmar digitalmente el mensaje. También se utilizan para generar claves a partir de frases; de este modo el usuario no tiene que recordar una clave compleja, ilegible y larga para que sea segura, sino una frase tan larga como quiera, que es transformada por un digest en su clave para descifrar.

Una huella digital es un conjunto de datos asociados a un mensaje que permiten asegurar que el mensaje no fue modificado. Esta huella digital o resumen se obtiene aplicando una función digest, a ese mensaje, esto da como resultado un conjunto de datos singular de longitud fija. [WWW 029]

Algoritmos Asimétricos más comunes

a) Algoritmo RSA

RSA es un Sistema Criptográfico de Claves Públicas y Autenticación que usa un algoritmo desarrollado en 1977 por Ron Rivest, Adi Shamir y Leonard Adleman., el algoritmo RSA es el más usado en Internet, es parte de los navegadores como Netscape e Internet Explorer, así como de muchos otros más. Es un algoritmo utilizado tanto para encriptar

información como para firma digital. Los sistemas de firma digital se utilizan para garantizar que el autor de la información es el firmante y no ha sido modificada por un tercero. La clave puede tener una longitud variable y puede ser tan grande como se desee y se pueda generar.

Características:

- ✓ Propósito: Encripción y Firma Digital.
- ✓ Rango de clave: 1024 bits para uso corporativo y 2048 para claves valuables.
- ✓ Fecha de Creación: 1977

En lugar de emplear una sola clave para encriptar y desencriptar datos, el sistema RSA emplea un par combinado de claves que desarrolla una transformación en un solo sentido. Cada clave es la función inversa de la otra, es decir, lo que una hace, sólo la otra puede deshacerlo. La Clave Pública en el sistema RSA es publicada por su propietario, en tanto que la Clave Privada es mantenida en secreto. Para enviar un mensaje privado, el emisor lo encripta con la Clave Pública del receptor deseado. Una vez que ha sido encriptado, el mensaje sólo puede ser descifrado con la Clave Privada del receptor. Inversamente, el usuario puede encriptar datos utilizando su Clave Privada. Es decir, las claves del sistema RSA pueden ser empleadas en cualquier dirección. Esto sienta las bases para la firma digital. Si un usuario puede desencriptar un mensaje con la Clave Pública de otro usuario, éste debe, necesariamente, haber utilizado su Clave Privada para encriptarlo originariamente. Desde el momento que solamente el propietario puede utilizar su propia Clave Privada, el mensaje encriptado se transforma en una especie de firma digital, un documento que nadie más ha podido crear. Bajo RSA se desarrolló el algoritmo estándar de firmas digitales para correos S/MIME. [WWW 030]

Funcionamiento.- Los números enteros (0, 1, 2... y sus opuestos -1,-2… etc.) tienen una estructura algebraica determinada con las operaciones producto y la suma. Esta estructura es la de anillo conmutativo y una de sus características es la existencia de un elemento neutro respecto al producto, que es la unidad. En este anillo existen dos divisores de la unidad (el número 1), el 1 y el -1.

Dados dos números enteros, p y q, es posible encontrar otros dos, c y r tales que p = q.c + r. A c se le suele llamar cociente y a r resto. Particularmente, existe un r tal que r < Iqi. Cuando r es cero, entonces decimos que q es un factor de p. Fijado un entero q, existen II restos posibles: 0, 1, 2,....,Iq-1I y es definible una relación de equivalencia: Dos enteros m y n son equivalentes si y sólo si m-n es un múltiplo de q. Esto es lo mismo que decir que tanto m como n tienen el mismo resto, o que m es congruente con n módulo q, y lo simbolizaremos por m = n (mod q). El conjunto de las clases de equivalencia forma a su vez un anillo y tendremos tantas como restos posibles.

Se dice que d es el máximo común divisor de dos números p y q cuando es el factor más grande de p y q: d = m.c.d (p,q). Dos números p y q son primos entre sí, cuando m.c.d (p,q) = 1. Un número p es primo cuando siempre que exista un factor q tal que p =q.k entonces k sólo se puede dividir por si mismo). Cualquier número q es un producto de primos y este producto es único (salvo divisores de la unidad). Existe un número infinito de primos, no hay una fórmula para obtenerlos y su distribución no se puede determinar por métodos numéricos.

b) **Algoritmo *DSA***

El Digital Signature Algorithm (DSA) fue publicado por el Instituto Nacional de Tecnología y Estándares (NIST) en el estándar llamado Digital Signature Standard (DSS) que es parte de gobierno de los Estados Unidos. DSS fue seleccionado por el NIST con ayuda del NSA (National Security Agency) para ser el estándar de autenticación digital del gobierno de los Estados Unidos a partir de Mayo 19 de 1994. DSA está basado en el problema de logaritmos discretos y se deriva de sistemas criptográficos propuestos por Schnorr y El Gamal. Es únicamente para autenticación.

Características:

- ✓ Propósito: Firmas Digitales
- ✓ Rango de clave: 56 bits
- ✓ Fecha de Creación: 1994

c) ***Diffle-Hellman (DH)***

Fue el primer algoritmo de clave pública inventado (1976). Tiene su seguridad en la dificultad de calcular logaritmos discretos infinitamente. DH se usa principalmente para distribución de claves, para generar claves secretas, aunque no es recomendable para encriptar ni desencriptar.

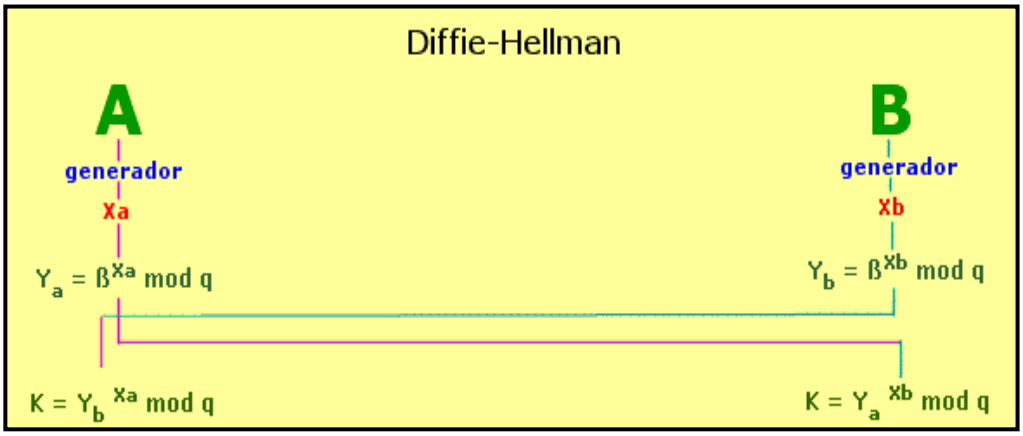


Figura 3.9 Diagrama del Algoritmo Diffie - Hellman

Su importancia se debe al hecho de ser el inicio de los sistemas asimétricos, ya que en la práctica sólo es válido para el intercambio de claves simétricas, y con esta funcionalidad es muy usado en los diferentes sistemas seguros implementados en Internet, como SSL (Secure Socket Layer) y VPN (Virtual Private Network).

Características:

- ✓ Propósito: Firmas Digitales.
- ✓ Rango de clave: 1536 bits.
- ✓ Fecha de Creación: 1976.

Funcionamiento.- DH basa su funcionamiento en las potencias de los números y en la función mod (módulo discreto). Uniendo estos dos conceptos se define la potencia discreta de un número como $Y = X^a \bmod q$. Si bien el cálculo de potencias discretas es fácil, la obtención de su función inversa, el logaritmo discreto, no tiene una solución analítica para números grandes.

Pasos de implementación:

1. Se busca un número primo muy grande, **q**.
2. Se obtiene el número **ß**, raiz primitiva de q, es decir, que cumple que $ß \bmod q$, $ß^2 \bmod q$,...., $ß^{q-1} \bmod q$ son números diferentes.
3. ß y q son las claves públicas.

Para generar una clave simétrica compartida entre dos usuarios, A y B, ambos parten de un generador de números pseudoaleatorios, que suministra un número de este tipo diferente a cada uno, X_a y X_b. Estos son las claves privadas de A y B. Con estos números y las claves públicas ß y q que ambos conocen, cada uno genera un número intermedio, Y_a e Y_b, mediante las fórmulas: $Y_a = ß^{Xa} \bmod q$, $Y_b = ß^{Xb} \bmod q$.

Estos números son intercambiados entre ambos, y luego cada uno opera con el que recibe del otro, obteniendo en el proceso el mismo número ambos: $K = Y_b^{Xa} \bmod q$, $K = Y_a^{Xb} \bmod q$

Este número K es la clave simétrica que a partir de ese momento ambos comparten, y que pueden usar para establecer una comunicación cifrada mediante cualquiera de los sistemas simétricos.

Con este esquema, si se desea compartir una clave privada con otro usuario cualquiera, basta con acceder a su Y_u y enviarle la nuestra. Para facilitar este proceso se suelen publicar las Y_u de todos los usuarios interesados en un directorio de acceso común. [WWW 030]

d) Algoritmo MD5

MD5 es una función de hashing de una sola dirección, produciendo un resultado de 128 bits. Después de un proceso inicial, MD5 procesa el texto insertado en bloques de 512 bits, divididos en 16 bloques de 32 bits. El resultado de el algoritmo son 4 bloques de 32 bits, que juntos forman un bloque de 128 bits.

Características:

- ✓ Propósito: Hashing (Digestión de documentos digitales)
- ✓ Rango de clave: 128 bits
- ✓ Fecha de Creación: 1992

e) El Gamal

Propuesto por T. El Gamal, este sistema de clave pública está basado en un procedimiento de cifrado que usa dos valores públicos: un número primo *p* de aproximadamente 200 dígitos y un entero *g* tal que sus potencias generan todos los elementos del grupo. Así, la clave secreta del firmante es un entero aleatorio *x* elegido por el mismo tal que $1 < x < p-1$, y la clave pública asociada *y* se obtiene como sigue: $y = g^x \pmod p$

El cifrado de un mensaje en claro M tal que 1 < M < p, se lleva a cabo eligiendo un valor entero aleatorio *k* con 1 < k < p-1 y *k* relativamente primo con *p*. Si los valores de *k* elegidos para la computación de un mismo mensaje en claro son distintos los cifrados resultantes también lo serán. Un inconveniente importante de este sistema de cifrado es la capacidad de almacenamiento necesaria, al ser la longitud del mensaje cifrado el doble que la del mensaje en claro.

La ruptura de este sistema pasa por la resolución de un problema de logaritmo discreto, lo cual resulta complicado cuando se trabaja con números suficientemente grandes. Sin embargo, en ocasiones el cálculo del logaritmo discreto resulta viable incluso para valores de *p* de gran tamaño, lo cual se debe a la existencia de números primos con características debilitantes para el sistema, esto es, números a partir de los que resulta posible obtener la clave secreta *x* a partir de la pública *y*, que deberemos evitar. Es el predecesor del DSS (Digital Signature Standard) y su uso está bastante extendido a pesar de que no se ha creado ningún estándar conocido para ello.

f) Algoritmo Rijndael

Para sustituir al ya obsoleto algoritmo DES, el NIST (National Institute of Standards and Technology) propuso una competición para desarrollar el estándar AES, hasta cuya resolución ha adoptado el sistema Triple-DES como una solución temporal. Los cinco algoritmos finalistas para AES, elegidos entre un total de quince, fueron MARS, RC6, Rijndael, Serpent y Twofish. Así, Rijndael es un cifrador en bloque diseñado por John Daemen y Vincent Rijmen como algoritmo candidato al AES (Advanced Encryption Standard). Su diseño estuvo fuertemente influenciado por el de un cifrador (block cipher Square), que también fue

creado por John Daemen y Vincent Rijmen y se centraba en el estudio de la resistencia al criptoanálisis diferencial y lineal. El nombre del algoritmo es una combinación de los nombres de sus dos creadores. El cifrador tiene longitudes de bloque y de clave variables y puede ser implementado de forma muy eficiente en una amplia gama de procesadores y mediante hardware. Como todos los candidatos del AES es muy seguro y hasta la fecha no se le han encontrado puntos débiles.

La longitud de la clave de Rijndael, debe ser de 128, 192 o 256 bits, según los requisitos establecidos para el AES. Asimismo, la longitud del bloque puede variar entre 128, 192 o 256 bits. Todas las posibles combinaciones (nueve en total) entre longitudes de clave y bloque son válidas, aunque la longitud oficial de bloque para AES es de 128 bits. Las longitudes de la clave y el bloque pueden ser fácilmente ampliadas a múltiplos de 32 bits. El número de iteraciones del algoritmo principal puede variar de 10 a 14 y depende del tamaño del bloque y de la longitud de la clave. Una de las críticas más habituales de Rijndael es el escaso número de iteraciones, pero esto no supone un problema, pues el coste operacional puede aumentarse sin más que incrementar el tamaño del bloque y la longitud de la clave.

La implementación Stealth de Rijndael usa una clave de 256 bits y un bloque de 128 bits de tamaño. Usando la mayor longitud posible de clave conseguimos la máxima seguridad para el usuario. La filosofía de este diseño concedería pues mayor importancia a la seguridad que a la velocidad. Si el usuario proporciona una clave de menor longitud Stealth la transforma de una forma especial, casi aleatoriamente, para hacerla de 256 bits. Y aunque acepta tamaños de bloque mayores que 128 bits, no

existe ninguna razón para usarlos siendo que este número de bits ha sido elegido como tamaño estándar.

g) **Algoritmo Basado en Curvas Elípticas**.

En 1985, la teoría de las curvas elípticas encontró de la mano de Miller aplicación en la criptografía. La razón fundamental que lo motivó fue que las curvas elípticas definidas sobre cuerpos finitos proporcionan grupos finitos abelianos, donde los cálculos se efectúan con la eficiencia que requiere un criptosistema, y donde el cálculo de logaritmos es aún más difícil que en los cuerpos finitos. Este algoritmo basa su seguridad en el Problema del Logaritmo Discreto Elíptico, es decir que basa su seguridad en la PLD sobre el grupo abeliano de puntos racionales de una curva elíptica sobre un campo finito.

La principal característica que tienen los sistemas sobre curvas elípticas es que el PLDE es totalmente exponencial, es decir no existe un algoritmo eficiente que calcule logaritmos discretos. Esto permite usar claves de longitud reducida en los
Sistemas criptográficos que los usan. Vale entonces realizar la siguiente comparación:

Claves CCE de 163b = claves RSA de 1024b
Claves CCE de 210b = claves RSA de 2048b

Otra característica de CCE es que su relativa longitud de la clave hace posible ser implementados en dispositivos de bajos recursos de procesamiento, memoria, ancho de banda,... (smart cards, ATM, teléfonos celulares, PCs,..) En la actualidad existen varios estándares que permiten el uso adecuado y óptimo de los CCE, entre los cuales se

encuentran: IEEE P1363 (Institute of Electrical and Electronics Engineers), el ANSI X9.62, 63, ANSI TG-17, ANSI X12 UN/EDIFACT, ISO/IEC 14888, ISO/IEC 9796-4, ISO/IEC 14946 (International Standards Organization), ATM Forum (Asynchronous Transport Mode), WAP (Wireles Application Protocol). En comercio electrónico: FSTC (Financial Services Technology Consortion), OTP 0.9 (Open Trading Protocol), SET. En internet IETF, IPSec.

Ataques a CCE

- ✓ Búsqueda a fuerza bruta calculando A, 2A, 3A, …
- ✓ Si el orden de la curva tiene todos los factores pequeños se puede aplicar el método de Pohlig-Hellman
- ✓ El método Baby-Step Giant-Step
- ✓ El método de la raíz de Pollard (pi*n/2)^(1/2)
- ✓ Método paralelizado de Pollard (pi*n)^1/2 / (2r)
- ✓ Ataque MOV (Menezes Okamoto Vanstone) que encaja una curva elíptica sobre un campo F_{q} a una curva elíptica sobre una extensión del campo F_{q^k}, si la curva es supersingular k<=6, por lo que no son recomendables para usos criptográficos hay que verificar si es necesario para el caso nosupersingular que n, el orden del punto base no divide a q^k -1 para los primeros k (hasta 20)
- ✓ No elegir curvas anómalas es decir que #E(F_{p}) = p
- ✓ No elegir curvas definidas sobre F_{2^m} con m compuesto, aunque no se ha mostrado una practicidad de este ataque cuando m es compuesto, algunos estándares recomiendan no usarlos

h) **Sistema Probabilístico**.

Aunque la criptografía de clave pública resuelve el importante problema de la distribución de claves que se presenta en la criptografía de clave secreta; en clave pública se presenta otro problema, el texto cifrado C=Ek (M) siempre deja escapar alguna información sobre el texto original porque el criptoanalista puede calcular por sí mismo la función de cifrado con la clave pública sobre cualquier texto que quiera. Dado cualquier M' de su elección, puede fácilmente descubrir si el mensaje

original M=M', pues esto se cumple si, y sólo si Ek(M')=C. Incluso aunque recuperar M a partir de C fuera efectivamente infactible, no sabemos cómo medir la información que deja escapar sobre M. El propósito de la criptografía probabilística (noción ideada por Golwaser y Micali) es cifrar mensajes de manera que no exista cálculo factible que pueda producir información en lo que respecta al texto original correspondiente (salvo con una probabilidad ínfima). Hay que decir que estos sistemas no ofrecen verdadero secreto perfecto, son totalmente inseguros contra criptoanalistas con poder de cálculo ilimitado.

La principal diferencia técnica entre el cifrado probabilístico y los criptosistemas de clave pública es que los algoritmos de cifrado son probabilísticos en lugar de determinísticos: el mismo mensaje original puede dar lugar a un gran número de criptogramas distintos. En consecuencia, un criptoanalista que tenga un candidato para el texto original no podría verificar su suposición cifrándolo y comparando el resultado con el criptograma interceptado.

Otros algoritmos de Encripción

- ✓ **Sistema de Rabin**. Se basa también en la factorización.
- ✓ **Sistema de Merkle-Hellman**. Esta basado en el problema de la mochila.
- ✓ **Sistema de McEliece**. Se basa en la teoría de la codificación algebraica, utilizando el hecho de que la decodificación de un código lineal general es un problema NP-completo.
- ✓ 3DES y ya se está implementando el AES (Advanced Encryption Standard).
- ✓ RC2
- ✓ RC4
- ✓ RC5
- ✓ ECC (Criptografía de Curvas Elípticas)

Otros algoritmos de Hashing

- ✓ **MD2.-** Message Digest 2. Se diseñó para ordenadores con procesador de 8 bits, y hoy apenas se utiliza. Se conocen ataques a versiones parciales de MD2. Es una función de un sentido usada en Privacy Enhanced Mail (PEM) junto con MD5. Produce código de hash de 128 bits para una entrada arbitraria. Es similar en su estructura a MD4 y MD5, pero más lento e inseguro
- ✓ **MD4.-** Message Digest 4. Fue desarrollado por Ron Rivest, de RSA Data Security. Su diseño es la base de otros hash, aunque se le considera inseguro. Un ataque desarrollado por Hans Dobbertin permite generar colisiones (mensajes aleatorios con los mismos valores de hash) en cuestión de minutos para cualquier PC. Por ese motivo, está en desuso. Se publicó inicialmente en 1990 y una versión revisada se publica como RFC 1320 en Abril del 992, junto con MD5. Comparte los objetivos de diseño con MD5, sin embargo MD5 es más complejo lo que lo hace más seguro, pero también más lento. Resumen es de 128 bits
- ✓ **SHA-1.-** SHA (Secure Hash Algorithm) fue desarrollado como parte del estándar hash seguro (Secure Hash Standard, SHS) y el estándar de cifrado digital (Digital Signature Standard, DSS) por la Agencia de Seguridad Nacional norteamericana (NSA). Aparentemente se trata de un algoritmo seguro y sin fisuras, al menos por ahora. La primera versión, conocida como SHA, fue mejorada como protección ante un tipo de ataque que nunca fue revelado. El documento FIPS (Federal Information Processing Standard) que oficialmente lo describe afirma que los principios subyacentes al SHA-1 son similares a los del MD4 de Rivest. Su implementación puede estar cubierta por patentes en Estados

Unidos y fuera de ellos. A falta de ataques ulteriores, se le puede considerar seguro. Es el algoritmo de firmado utilizado por el programa PGP en sus nuevas claves DH/DSS (que significa: cifrado mediante clave Diffie-Hellman y firmado mediante función hash/ Digital Signature Standard).

Para la generación de otro tipo de firmas digitales suelen usarse algoritmos basados en criptografía de clave pública, sobre todo RSA y DSS. [WWW 031]

3.6 Criptografía y el web

En todo momento cuando se necesita enviar y recibir información por la WWW, realizar algún tipo de transacción, una compra en línea, consultar el saldo de nuestra cuenta en el banco, etc. la preocupación más importante es saber si realmente todas estas transacciones están realizándose de una manera segura de tal manera que nada ni nadie pueda interceptar los datos proporcionados en las transacciones. Todos queremos estar protegidos contra ataques a nuestra información sobre todo de los **hackers**, es por esto que la mejor solución hasta el momento ha sido utilizar la Criptografía como único mecanismo de seguridad para la información que se envía y recibe a través del Internet. La Criptografía se ha convertido en una tecnología fundamental para proteger la información, sin embargo se requiere de muchos recursos tanto técnicos como humanos y sobre todo tecnológicos y económicos para asegurar las comunicaciones de una empresa u organización.

Al existir muchas técnicas criptográficas que cubren distintas necesidades, se hace más necesario que dentro de cada organización exista la suficiente capacidad para tomar decisiones, las más certeras que permitan el aseguramiento de la información, en muchos casos las

diferencias existentes entre los sistemas de encriptación son técnicas, en otros casos las diferencias son resultados de restricciones con lo que tiene que ver con aspectos legales como patentes, secretos comerciales, etc. y lo más común, sobre todo en nuestros países subdesarrollados, las restricciones criptográficas son resultado de decisiones políticas.

La Criptografía y la Seguridad en la Web

Para la mayoría de expertos en seguridad electrónica, se han identificado cuatro puntos clave que se debe cumplir para describir todas las funciones que tiene la encriptación en los sistemas de información modernos:

a. **Confidencialidad.-** la encriptación se utiliza para ocultar la información a través de Internet y almacenarla en servidores de manera que cualquiera que intente interceptar no pueda tener acceso al contenido de los datos. Para muchos esta propiedad es "privacía", aunque para la mayoría simplemente es protección de la información de la agregación o el uso inapropiado.
b. **Autenticación.-** las firmas digitales sirven para identificar al autor del mensaje, las personas que reciben el mensaje pueden comprobar la identidad de quién lo firmó, pueden utilizarse junto con claves de acceso o como alternativa a las claves.
c. **Integridad.-** para verificar que un mensaje no ha sido modificado durante el camino que recorre desde su emisor hasta su receptor, se pueden utilizar varios métodos, mediante códigos de Compendios de Mensajes firmados digitalmente.
d. **No Repudio.-** mediante la encriptación se crean recibos de forma que el autor de un mensaje no pueda negar falsamente su envío.

Problemas a los que la Criptografía no da una solución adecuada

- ✓ **Protección de documentos no encriptados.-** aun cuando se configuren los servidores Web para que sólo envíen archivos a conexiones que utilicen SSL, siempre los originales sin encriptar permanecerán en el servidor, a menos que se encripten los documentos independientemente, siguen siendo vulnerables si alguien viola el servidor y tiene acceso a la información.
- ✓ **Protección contra el robo de llaves de encriptación.-** es hacer posible que quienes tienen las llaves criptográficas puedan desencriptar los archivos o mensajes, por ello cualquier atacante que pueda robar o comprar una llave podrá desencriptar cualquier archivo o mensaje encriptado con dicha llave. El principal problema es que SSL permite tener copias de la llave secreta del servidor en el disco duro de la computadora.
- ✓ **Protección contra ataques de negación del servicio.**- muchos protocolos criptográficos como SSL dan protección segura contra la intercepción de la información, pero los atacantes informáticos tienen muchos otros propósitos, no simplemente el daño se lo hace interceptando la información sino dañando los sistemas de comunicaciones o accediendo a los servidores y borrando la información.
- ✓ **Protección contra programas de encriptación con trampas.-** un atacante puede acceder y modificar un programa de encriptación para hacerlo fraudulento, puede hacer que el navegador Web por defecto del servidor utilice la misma llave de encriptación y así podrá cometer el delito. Ante esto lo más prudente será obtener sistemas de encriptación confiables y cuando se obtenga algún software solicitar su correspondiente

Firma de Código para así poder detectar cambios en la programación de los mismos.

- ✓ **Protección contra errores.-** la información nunca podrá estar segura si después de transmitirle en forma encriptada, el receptor no toma las debidas precauciones al desencriptar y usa dicha información de una forma maliciosa, más aun si una persona encargada de la administración de los sistemas informáticos es sorprendida por alguna persona que aduce ser miembro de la policía, de seguridad, etc. y hace que le proporcione las claves de acceso.

3.7 Estándares criptográficos y electrónicos aplicables

Las políticas de certificación y prácticas establecen un marco de estandarización de las actividades que permite la operación efectiva de la firma electrónica desde el punto de vista técnico. A continuación se mencionan algunos documentos de referencia que están siendo utilizados como estándares de la industria de certificación digital y PKI. Estos documentos pueden agruparse, según su nivel de operación, en los siguientes temas:

ESTANDARES INTERNACIONALES

- ✓ **ISO 9796,** Organización de Estándares Internacionales ("International Standards Organization "), Norma ISO 9796 de Tecnología de la Información, Técnicas de Seguridad, Mecanismo de Firma Digital ("Information Technology - Security Techniques - Digital Signature Scheme"). [WWW 032]
- ✓ **ANSI X9.31,** Instituto Americano de Estándares Nacionales ("American National Standards Institute"), estándar X9.31 de Autenticado de Mensajes para Instituciones Financieras "Financial

Institution Message Authentication" para el sistema bancario estadounidense

- ✓ **ITU-T X.509,** Unión Internacional de Telecomunicaciones, Sector de Estandarización de Teleco-municaciones ("International Telecommunication Union, Telecommunication Standardization Sector"), estándares X.509 de Tecnología de la Información – Interconexión de Sistemas Abiertos – El Directorio: Marco para el Autenticado ("Information Technology - Open Systems Interconnection - The Directory: Authentication Framework") [WWW 033]
- ✓ **PKCS,** Estándares de Criptografía de Clave Pública ("Public Key Cryptography Standards") desarrollados por RSA Corporation en forma conjunta con Apple, Microsoft, Digital, Lotus, Sun y Massachussets Institute of Technology. [WWW 034]
- ✓ **SWIFT,** Sociedad para las Telecomunicaciones Financieras Interbancarias Mundiales ("Society for Worldwide Interbank Financial Telecommunications")

Declaración de Prácticas y Política de Certificación

- ✓ ANSI X9.79: Public Key Infraestructure (PKI), Practices and Policy Framework.
- ✓ RFC25272: Internet X.509, Public Key Infraestruture, Certificate Policy and Certification Practices Framework.

Seguridad

- ✓ ISO/IEC 17799:2000 Information Technology - Code for information security management.

- ✓ BS 7799 Part 2 The specification for information security management systems, 1998, en la cual se basó la Norma ISO/IEC 17799.
- ✓ ISO IS 15408 Common criteria version 2.1 (2000)
- ✓ FIPS PUB 140-1. Security requirements for cryptography modules, October 2001.

Estructura de Certificados

- ✓ ITU-T X.509
- ✓ ISO/IEC 9594

Repositorio de Información para implementación de PKI

- ✓ RFC 1777 Yeong, W. Et al., Lightweight Directory Access Protocol, Marzo 1995.
- ✓ RFC 2251 Wahl, M. Et al. Lightweight Directory Access Protocol v3, Diciembre 1997.
- ✓ RFC 2559 Boeyen, S. Et al. Internet X.509 Public Key Infraestructure. Operational Protocols – LDAPv2, Abril 1999.
- ✓ RFC 2585 Housley, R., Hoffman, P., X.509 Internet Public Key Infraestructure
- ✓ Operational Protocols: FTP and HTTP, Myo 1999.
- ✓ RFC 2587 Boeyen, S. Et al., Internet X.509 Public Key Infraestructure. LDAPv2 Schema, Junio 1999.
- ✓ X.500 ITU-T Recommendation X.500, Information technology – Open Interconnection – The Directory: Overview of concepts, models and services, 1997.

Organismos de Estandarización

- ✓ [IEC] International Engineering Consortium. [WWW 035]
- ✓ [IETF] The Internet Engineering Task Force. [WWW 036]

- ✓ [ISO] International Organization for Standardization. [WWW 032]
- ✓ [ITU] International Telecommunication Union. [WWW 037]
- ✓ [ITU-T] ITU Telecommunication Standardization Sector.
- ✓ [PKIX] Public-Key Infrastructure (X_509). [WWW 038]
- ✓ [SMIME] S/MIME Mail Security (SMIME). [WWW 039]
- ✓ [STD 1] ITU-T Recommendation X.680 | ISO/IEC 8824-1, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", 1997

Estándar de Notación Abstracta

- ✓ [ISO/IEC 8824-1] ITU-T Recommendation X.680 | ISO/IEC 8824-1, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", 1997
- ✓ [ISO/IEC 8825-1] ITU-T Recommendation X.690 | ISO/IEC 8825-1, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", 1997
- ✓ [X.208] CCITT Recommendation X.208, "Specification of Abstract Syntax Notation One (ASN.1)", 1988

Certificación Electrónica

- ✓ [RFC 2459] Housley, R, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Enero 1999. [WWW 040]
- ✓ [RFC 2510] Adams, C., Farrel, S., "Internet X.509 Public Key Infrastructure. Certificate Management Protocols", Marzo 1999. [WWW 041]

- ✓ [X.509v3] ITU-T Recommendation X.509 | ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Authentication framework", 1997
- ✓ [X.509v4] Draft Revised ITU-T Recommendation X.509 | ISO/IEC 9594-8, "Information technology - Open Systems Interconnection - The Directory: Public Key and Attribute Certificate Frameworks", 2000 (work in progress)

Terceras Partes de Confianza

- ✓ ISO/IEC 10181-4, "Information Technology - Open Systems Interconnection - Security frameworks for open systems: Non-repudiation framework", 1997
- ✓ ISO/IEC-18014, "Information Technology - Security Techniques - Time Stamping Services". Working Document. Enero 2000.
- ✓ [RFC 2560] Myers, M. et al., "X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol - OCSP", Junio 1999. [WWW 042]

Estándares de Criptografia de Clave Pública

- ✓ [PKCS#1] RSA Laboratories, "PKCS#1: RSA Cryptography Standard, Version 2.0", Octubre 1998
- ✓ [PKCS#3] RSA Laboratories, "PKCS#3: Diffie-Hellman Key Agreement Standard, Version 1.4", Noviembre 1993
- ✓ [PKCS#5] RSA Laboratories, "PKCS#5: Password-Based Cryptography Standard, Version 2.0", Marzo 1999
- ✓ [PKCS#6] RSA Laboratories, "PKCS#6: Extended-Certificate Syntax Standard, Version 1.5", Noviembre 1993
- ✓ [PKCS#7] RSA Laboratories, "PKCS#7: Cryptographic Message Syntax, Version 1.5", Noviembre 1993

- ✓ [PKCS#8] RSA Laboratories, "PKCS#8: Private-Key Information Syntax Standard, Version 1.2", Noviembre 1993
- ✓ [PKCS#9] RSA Laboratories, "PKCS#9: Selected Object Classes and Attribute Types, Version 2.0", Febrero 2000
- ✓ [PKCS#10] RSA Laboratories, "PKCS#10: Certification Request Syntax Standard, Version 1.7", Mayo 2000
- ✓ [PKCS#11] RSA Laboratories, "PKCS#11: Cryptographic Token Interface Standard, Version 2.10", Diciembre 1999
- ✓ [PKCS#12] RSA Laboratories, "PCKS#12: Personal Information Exchange Syntax, Version 1.0", Junio 1999
- ✓ [PKCS#15] RSA Laboratories, "PKCS#15: Cryptographic Token Information Syntax Standard, Version 1.1", Junio 2000
- ✓ [RFC 2314] Kaliski, B., "PKCS #10: Certification Request Syntax, Version 1.5.", RFC 2314, Marzo 1998. [WWW 043]
- ✓ [RFC 2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax, Version 1.5.", RFC 2315, Marzo 1998. [WWW 044]
- ✓ [RFC 2437] Kaliski, B., Staddon, J., "PKCS #1: RSA Cryptography Specifications, Version 2.0.", RFC 2437, Octubre 1998. [WWW 045].

Seguridad en MIME

- ✓ [RFC 2311] Dusse, S. et al., "S/MIME Version 2. Message Specification", Marzo 1998. [WWW 046]
- ✓ [RFC 2312] Dusse, S. et al., "S/MIME Version 2. Certificate Handling", Marzo 1998. [WWW 047]
- ✓ [RFC 2632] Network Working Group, "S/MIME Version 3. Certificate Handling", Junio 1999. [WWW 048]
- ✓ [RFC 2633] Network Working Group, "S/MIME Version 3. Message Specification", Junio 1999. [WWW 049]

- ✓ [RFC 2634] Network Working Group, "Enhanced Security Services for S/MIME", Junio 1999. [WWW 050]

Sintaxis de Mensaje Criptográfico

- ✓ [PKCS#7] RSA Laboratories, "PKCS#7: Cryptographic Message Syntax, Version 1.5", Noviembre 1993
- ✓ [RFC 2630] Housley, R., "Cryptographic Message Syntax", Junio 1999. [WWW 051]

Mecanismos de Autentificación

- ✓ [RFC 2025] Adams, C; "The Simple Public-Key GSS-API Mechanism (SPKM)", Octubre 1996. [WWW 052]
- ✓ [SSL 2] Netscape Communications Corp., "the SSL Protocol" Febrero 1995
- ✓ [SSL 3] Netscape Communications Corp., "the SSL 3.0 Protocol" Noviembre 1996
- ✓ [RFC 2743] Linn, J., "Generic Security Service Application Program Interface. Version 2, Update 1", Enero 2000. [WWW 053]
- ✓ [RFC 2246] T.Dierks et al., "The TLS Protocol. Versin 1.0"_Enero 1999 [WWW 054]
- ✓ OSI [X.200]/[ISO/IEC 7498-1] ITU-T Recommendation X.200, "Information technology - Open Systems Interconnection - Basic reference model: The basic model", 1994

Autoridades de Certificación Internacionales

- ✓ **Argentina** [WWW 055]
- ✓ **Francia** [WWW 056]
- ✓ **Italia** [WWW 057]
- ✓ **Reino Unido**, entre otras.

3.8 Comprobación de Integridad de Mensajes

Mecanismos de seguridad

- ✓ **Intercambio de autenticación:** garantiza que una entidad, ya sea origen o destino de la información, es la más segura y deseada.

- ✓ **Cifrado:** garantiza que la información no es inteligible para individuos, entidades o procesos no autorizados (confidencialidad). Consiste en transformar un texto en claro mediante un proceso de cifrado en un texto cifrado, gracias a una información secreta o clave de cifrado.

- ✓ **Integridad de datos:** implica el cifrado de una cadena comprimida de datos a transmitir, llamada valor de comprobación de integridad (Integrity Check Value). Este mensaje se envía al receptor junto con los datos ordinarios. El receptor repite la compresión y el cifrado posterior de los datos y compara el resultado obtenido con el que le llega, para verificar que los datos no han sido modificados.

- ✓ **Firma digital:** este mecanismo implica el cifrado, por medio de la clave secreta del emisor, de una cadena comprimida de datos que se va a transferir. La firma digital se envía junto con los datos ordinarios. Este mensaje se procesa en el receptor, para verificar su integridad. Este mecanismo es esencial en el servicio de no repudio.

- ✓ **Control de acceso:** sólo aquellos usuarios autorizados pueden acceder a los recursos del sistema o a la red, mediante claves de acceso, autorizaciones, etc.

- ✓ **Tráfico de relleno:** consiste en enviar tráfico espurio junto con los datos válidos para que el atacante no sepa si se está enviando información, ni qué cantidad de datos útiles se está transmitiendo.

- ✓ **Control de encaminamiento:** permite enviar determinada información por determinadas zonas consideradas clasificadas. Asimismo posibilita solicitar otras rutas, en caso que se detecten persistentes violaciones de integridad en una ruta determinada.

- ✓ **Unicidad:** consiste en añadir a los datos un número de secuencia, la fecha y hora, un número aleatorio, o alguna combinación de los anteriores, que se incluyen en la firma digital o integridad de datos. De esta forma se evitan amenazas como la reactuación o resecuenciación de mensajes.

Los mecanismos básicos pueden agruparse de varias formas para proporcionar los servicios previamente mencionados. Conviene resaltar que los mecanismos poseen tres componentes principales:

- ✓ Una información secreta, como claves y contraseñas, conocidas por las entidades autorizadas.

- ✓ Un conjunto de algoritmos, para llevar a cabo el cifrado, descifrado, hash y generación de números aleatorios.

- ✓ Un conjunto de procedimientos, que definen cómo se usarán los algoritmos, quién envía qué a quién y cuándo.

Los sistemas de seguridad requieren una gestión de seguridad, que comprende dos campos bien amplios:

- ✓ Seguridad en la generación, localización y distribución de la información secreta, de modo que sólo pueda ser accedida por aquellas entidades autorizadas.

- ✓ La política de los servicios y mecanismos de seguridad para detectar infracciones de seguridad y emprender acciones correctivas. [LIB 01]

3.9 Escuchas Electrónicas

Las escuchas electrónicas son una de las formas de delitos informáticos que se utilizan y emplean para el robo de la información, los "delitos electrónicos", son cualquier conducta criminal que en su realización hace uso de la tecnología electrónica ya sea como método, medio o fin.

Existen varios delitos que con diversos nombres realizan diferentes acciones que conducen al fraude informático, se pueden citar a:

- ✓ **La "bomba lógica".-** es la alteración de un programa con la finalidad de detener el funcionamiento del sistema en el momento decidido por el autor del hecho, destruir los datos o los programas de los mismos.
- ✓ **El virus informático.-** programa que pasa de mano en mano entre los usuarios, produciéndose el contagio entre los equipos informáticos con la consecuente destrucción de todos o parte de los sistemas con los que opera al ingresarse una determinada instrucción o en un tiempo dado.

Las características principales de este tipo de delitos son:

- ✓ Conductas criminales de cuello blanco, en tanto que sólo un determinado número de personas con ciertos conocimientos (en este caso técnicos) pueden llegar a cometerlas.
- ✓ Son acciones ocupacionales, que se realizan cuando la persona se halla trabajando.

- ✓ Son acciones de oportunidad, que aprovechan una ocasión creada o altamente intensificada en el mundo de funciones y organizaciones del sistema tecnológico y económico.
- ✓ Provocan serias pérdidas económicas, pues casi siempre producen "beneficios" de más de cinco cifras a aquellos que las realizan.
- ✓ Ofrecen posibilidades de tiempo y espacio, ya que en milésimas de segundo y sin una necesaria presencia física pueden llegar a consumarse.
- ✓ Son muchos los casos y pocas las denuncias, y todo ello debido a la misma falta de regulación por parte del Derecho.
- ✓ Son muy sofisticados y relativamente frecuentes en el ámbito militar.
- ✓ Presentan grandes dificultades para su comprobación, por su mismo carácter técnico.
- ✓ En su mayoría son imprudenciales y no necesariamente se cometen con intención.
- ✓ Ofrecen facilidades para su comisión a los menores de edad.
- ✓ Tienden a proliferar cada vez más, por lo que requieren una urgente regulación.

Los delitos como instrumento

- ✓ Falsificación de documentos vía computarizada (tarjetas de crédito, cheques, etc.)
- ✓ Variación de los activos y pasivos en la situación contable de las empresas.
- ✓ Planeamiento y simulación de delitos convencionales (robo, homicidio, fraude, etc.)
- ✓ Lectura, sustracción o copia de información confidencial.
- ✓ Modificación de datos tanto en la entrada como en la salida.

- ✓ Aprovechamiento indebido o violación de un código para penetrar a un sistema introduciendo instrucciones inapropiadas.
- ✓ Variación en cuanto al destino de pequeñas cantidades de dinero hacia una cuenta bancaria apócrifa.
- ✓ Uso no autorizado de programas de cómputo.
- ✓ Alteración en el funcionamiento de los sistemas, a través de los virus informáticos.
- ✓ Obtención de información residual impresa en papel luego de la ejecución de trabajos.
- ✓ Acceso a áreas informatizadas en forma no autorizada.

Los delitos como fin u objetivo.

- ✓ Programación de instrucciones que producen un bloqueo total al sistema.
- ✓ Destrucción de programas por cualquier método.
- ✓ Daño a la memoria.
- ✓ Atentado físico contra la máquina o sus accesorios.
- ✓ Sabotaje político o terrorismo en que se destruya o surja un apoderamiento de los centros neurálgicos computarizados.
- ✓ Secuestro de soportes magnéticos entre los que figure información valiosa con fines de chantaje (pago de rescate, etc.).

Existen tipos de delito que son cometidos y que se encuentran ligados directamente a acciones efectuadas contra los propios sistemas como son:

- ✓ Acceso no autorizado: Uso ilegitimo de passwords y la entrada de un sistema informático sin la autorización del propietario.
- ✓ Destrucción de datos: Los daños causados en la red mediante la introducción de virus, bombas lógicas, etc.

- ✓ Infracción al copyright de bases de datos: Uso no autorizado de información almacenada en una base de datos.
- ✓ Interceptación de e-mail: Lectura de un mensaje electrónico ajeno.
- ✓ Estafas electrónicas: A través de compras realizadas haciendo uso de la red.
- ✓ Transferencias de fondos: Engaños en la realización de este tipo de transacciones.

Por otro lado, la red Internet permite dar soporte para la comisión de otro tipo de delitos:

- ✓ Espionaje: Acceso no autorizado a sistemas informáticos gubernamentales y de grandes empresas e interceptación de correos electrónicos.
- ✓ Terrorismo: Mensajes anónimos aprovechados por grupos terroristas para remitirse consignas y planes de actuación a nivel internacional.
- ✓ Narcotráfico: Transmisión de fórmulas para la fabricación de estupefacientes, para el blanqueo de dinero y para la coordinación de entregas y recogidas.
- ✓ Otros delitos: Las mismas ventajas que encuentran en la Internet los narcotraficantes pueden ser aprovechadas para la planificación de otros delitos como el tráfico de armas, proselitismo de sectas, propaganda de grupos extremistas, y cualquier otro delito que pueda ser trasladado de la vida real al ciberespacio o al revés". [WWW 059]

3.10 Crackers y Hackers

Hackers.- 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 sus inteligencias, hasta aquellos cuyo único objetivo es sabotear una red, llevándose toda la información que posea para luego venderla. Los Hackers se consideran a si mismos una casta, y su filosofía de la vida es casi una religión. Delincuentes para unos, Héroes para otros, hoy los *hackers* están considerados por muchos como los dominantes de la era Tecnológica. 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 actividades de los *crackers* y *phreakers* suelen ser siempre delictivas, mientras que las de los *hackers* en algunos casos no lo son. Claro que todos ellos justifican sus formas de pensar y actuar con argumentos de lo más valederos para ellos, 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.

Un Hacker es un individuo muy ingenioso y bien informado muy conocedor de las técnicas Criptográficas, que se dedica a buscar y explotar fallos más o menos sutiles en los sistemas de seguridad. Puesto que cada sistema es único, los buenos Hackers suelen elaborar ataques *a medida*, poniendo a prueba su profundo conocimiento sobre las redes de ordenadores. Es prácticamente imposible proteger un sistema al cien por cien de un ataque de esta naturaleza. La actuación de los Hackers es muy

variada, muchos actúan solos, otros suelen formar grupos, en los que cada uno tiene su Alias[44], y que normalmente sólo establecen contacto a través de la red, no conociéndose entre ellos.

Cómo actúa un *Hacker*

Para infiltrarse en computadores ajenos a través de la Red, TCP/IP es el protocolo que se ha impuesto como norma universal *de facto* en las comunicaciones.

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

Tabla 3.1 Algunos puertos TCP/IP.

Protocolo TCP/IP. Demonios y Puertos

Internet basa su funcionamiento en el protocolo TCP/IP, y aunque existan otros protocolos para redes locales, la mayoría de sistemas operativos actuales permiten su coexistencia, por lo que cualquier máquina conectada a Internet *entiende* el protocolo TCP/IP. Un computador con TCP/IP puede establecer múltiples comunicaciones simultáneamente, a través de los denominados *puertos bien conocidos*. Un puerto se comporta como los canales de un televisor: a través de un único cable llegan múltiples emisiones, de las cuales podemos escoger cuál ver con solo seleccionar el canal correspondiente. Existen puertos dedicados a tareas concretas. Así por ejemplo el puerto 80 se emplea

[44] Nombre ficticio por el cual es reconocido entre su grupo.

para las páginas *Web*, y el 21 para la transferencia de ficheros. En la tabla 3.1 podemos ver algunos de los más usuales, aunque existen muchos más. Nada nos impedirá situar nuestro demonio de FTP en el puerto 300, por ejemplo, aunque eso obligará 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*[45] es un programa 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. 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

[45] Programa que cuando se ejecuta, espera que algún suceso lo active.

ser que se les diga lo contrario, abriendo inevitablemente agujeros de seguridad. [LIB 02]

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 el. 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. 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.

Borrando las 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 Hacker debe eliminar todas las entradas de dicho registro relativas a su paso por el sistema, si no quiere verse en problemas 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, este sea realmente confuso. Para ello nada mejor que emplear otros ordenadores más

modestos como plataforma para atacar al auténtico objetivo. Normalmente los Hackers buscan ordenadores poco protegidos, entran en ellos, y controlándolos remotamente intentan encontrar las debilidades del objetivo real. Esta estrategia haría que en los registros del sistema atacado aparezcan datos sospechosos acerca del ordenador intermedio, pero pocas veces del auténtico enemigo. Las autoridades tendrían que ponerse en contacto con el ordenador empleado como plataforma para buscar el indicio del verdadero atacante.

Cómo Protegerse del Ataque de los *Hackers*

- ✓ Almacene en su ordenador sólo la Información Necesaria. no almacene información sensible en su ordenador si esta no necesita ser consultada desde el exterior.
- ✓ 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.