Universidad Nacional Autónoma de Nicaragua
Unan - León

Facultad de Ciencias

Departamento de Computación

" Criptografía Moderna"

(CriptoSistemas de Clave Pública y Privada)

Trabajo de Monografía previo para optar al título de
Licenciado en Computación.

Integrantes:

- *Br. Omar Cornelio Morán Torres.*
- *Br. Rafael Antonio Poveda Moreno.*
- *Br. Luis Benito Quintero Cadena.*

Tutor:

MSc. Martín Ibarra Padilla.

Noviembre, 2003.

DEDICATORIA

No queremos dar partida a nuestro trabajo de fin de carrera sin antes agradecer el haber llegado al final de nuestro proyecto, de primera instancia a ***Dios*** *ya que iluminó nuestro camino para que todo el trabajo llegase con buena luz hasta el final, además aquellas personas que contribuyeron de manera directa como lo fueron nuestros padres quienes nos proporcionaron todo el apoyo moral y económico, nuestro tutor, Msc. Martín Ibarra Padilla quien nos brindó su apoyo y nos sirvió de guía para presentar una buena tesis; y por último también aquellas personas que nos apoyaron indirectamente como lo fue el D. Jorge Ramió, profesor titular de la Universidad Politécnica de Madrid, España quien nos ayudó de tal manera que nuestro proyecto tuviese gran valor ejemplar.*

ANTECEDENTES

El uso que ha venido imponiendo la ***Criptografía*** desde los siglos anteriores para una comunicación segura.
El cambio que ha sufrido a finales del siglo pasado, ya que entra a formar parte importante dentro la complejidad computacional.

Sabemos que no existe un sistema computarizado que garantice al 100% la seguridad de la información, debido a la inmensa mayoría de formas diferentes con que se puede romper la seguridad en un sistema.

La ***Criptografía Simétrica y Asimétrica*** en conjunto con otras técnicas, como el buen manejo de las claves y la legislación adecuada, resuelven satisfactoriamente los problemas de seguridad de la información.

OBJETIVOS GENERALES

- Dar a conocer las **Herramientas de Seguridad** Informática mas comunes, tratando de enfatizar la importancia del uso de la ***Criptografía Moderna.***

- Dar a conocer y aplicar algunos algoritmos mas importantes que se utilizan en la ***Criptografía Moderna*** para seguridad a la información a través de un manual.

OBJETIVOS ESPECIFICOS

- Dar el significado conceptual de todos los elementos básicos que se utilizan en el uso de la ***Criptografía Moderna,*** contenidos en el manual.

- Dar a conocer los algoritmos de **Clave Simétrica** y de **Clave Pública** mas usuales para la protección de la información.

- Citar algunas técnicas de prevención o **Mecanismos de Seguridad** que se necesita en un Sistema Computarizado para evitar posibles ataques informativos.

- Utilizar una aplicación o software para ejemplificar algunos de los algoritmos planteados en el manual para la protección de la información.

RESUMEN DEL CONTENIDO

CAPÍTULO 1: ***CONCEPTOS GENERALES***

CAPÍTULO 2: ***HERRAMIENTAS DE SEGURIDAD***

CAPÍTULO 3: ***CRIPTOSISTEMAS MODERNOS***

CAPÍTULO 4: ***APLICACIONES DE SEGURIDAD***

CAPÍTULO 5: ***CERTIFICADOS DIGITALES***

CAPÍTULO 6: ***OTRAS HERRAMIENTAS CRIPTOGRÁFICAS***

CAPÍTULO 7: ***USO DE LA APLICACIÓN***

CONTENIDO

PREFACIO

Antes de comenzar con el desarrollo de nuestro tema, quizás sea importante mencionar algunos datos estadísticos relacionados con la **"Seguridad la Información"** en algunas de las empresas más importantes alrededor del mundo.

Uno de los reportes de **"Computer Crime Survey"** del **FBI**, proporcionado por Secure Site E-News del 22 de mayo de 1999, de la compañía **VeriSing**, se dieron los siguientes datos:

Se estudiaron 521 compañías de varias ramas de la Industria y de diferentes tamaños. Estas están actualmente trabajando para que su sistema computarizado sea seguro.

El 94% de las organizaciones tiene actualmente un sitio en la web.
El 61% de estas compañías u organizaciones ha tenido experiencias de pérdidas debido al uso no autorizado de su sistema computarizado.
El 50% de todas las compañías reportaron abuso del uso de la red.
El 32% de estas organizaciones están usando ahora métodos de identificación segura en su sitio de Internet.
El promedio de pérdida de robo o pérdida de información está sobre $1.2 millones de dólares.
El promedio de pérdida por sabotaje esté sobre $1.1 millones de dólares.

A la pregunta**, ¿Qué tipo de tecnología de seguridad usa?**, Se contestó con lo siguiente:
- El 89% cuenta con un Control en el Acceso al Sistema.
- El 88% usa Firewalls.
- El 59% cuenta con Archivos Cifrados.
- El 59% cuenta con un Sistema de Passwords.
- El 44% usa Sistema de Log-in cifrados.
- El 40% utiliza la Detección de intrusos.
- El 37% usa Smart-Cards.
- El 32% Certificados Digitales para la Autenticación.

A la pregunta**, ¿Cuál es el más frecuente orígen de un ataque?.**
- Un 86% de las compañías respondió: un empleado disgustado.
- Un 74% de las compañías respondió: un hacker independiente.
- Un 53% de las compañías respondió: Un competidor.

A la pregunta, **¿Su organización provee servicio de comercio electrónico?**
- El 29%, dijo sí.

A la pregunta, **¿Su web-site ha tenido un acceso no autorizado en los últimos 12 meses?**

- Un 44% respondió: No.
- Un 38% respondió: No sabe.
- Un 18% respondió: Sí.

Enseguida damos a conocer uno de los reportes que se tiene sobre el tema de Seguridad en la Información en Europa, dado a conocer en unos cursos de ***Criptografía Industrial*** en Bélgica en junio de 1997; en donde se mide la frecuencia de incidentes de seguridad de la información relacionada con sus causas.

Frecuencia	*Razón*
50 – 60%	Errores debido a la importancia, reacciones de pánico, mal uso...
15 – 20%	Empleados disgustados, accidentes de mantenimiento.
10 – 15%	Desastres naturales como inundaciones, incendios
3 – 5%	Causas externas: "hacker"

Figura 1.

Otro aspecto de gran importancia a considerar en este amplio tema, es el uso de la ***Criptografía*** en **Internet**.

Básicamente toda la estructura del **"Comercio Electrónico"**, dependerá de ella. Tanto es así, que en años anteriores los volúmenes de las transacciones en línea no superaban los 500 millones de dólares por año. Las perspectivas más modestas declaraban que esa cifra subiría hasta los 22.000 millones de dólares para el final del milenio. Nicholas Negroponte, uno de los más reconocidos gurús de **Internet,** considera montos por cien mil millones de dólares.

Además la **Internet** a crecido y a tenido un gran desarrollo; para muestra algunos datos proporcionados por Van Oorschot de Entrust Technologies en una conferencia del ciclo, "The Mathematics of Public Key Cryptography", en junio de 1999, son los siguientes:

- El tráfico de Internet se duplica cada 100 días.
- En enero de 1999 hubo 150 millones de personas en línea conectadas a la Internet, 75 millones de ellas en los Estados Unidos.
- El comercio en la Internet se duplica cada año.
- Para el año 2002 se dedujo que lo comercializado en Internet llegó a $1 trillón de dólares.
- A la radio le tomó 40 años, a la televisión 10 años para alcanzar 50 millones de usuarios a la red (Internet), le ha tomado menos de 5 años.

Estos datos solamente son algunos de los que frecuentemente son dados a conocer por algún medio, y aunque la mayoría obedecen a intereses comerciales, lo que sí es verdadero es el enorme cambio que han tenido gran cantidad de actividades a raíz del uso de **Internet**, que incluso, se ha considerado como el invento más importante de fin de siglo y de ahí lo primordial de todo lo relacionado con su seguridad.

Para reafirmar la trascendencia o importancia que tiene la Seguridad en los Sistemas Computarizados, siempre podremos encontrar razones, por medio de las cuales podemos atacar este problema, de la falta de Seguridad en los Sistemas Computarizados.

El diseñar una estrategia de seguridad depende en gran parte de la actividad que se este desarrollando, sin embargo se puede considerar los siguientes tres pasos generales:

- **Política Global de Seguridad.**

  - Se debe de establecer el status de la información para la empresa u organización.
  - Debe de contener un objetivo general.
  - La importancia de la tecnología de la información para la empresa.
  - El período de tiempo de validez de la política.
  - Los recursos con que se cuenta.
  - Objetivos específicos de la empresa.

  Debe establecerse la calidad de la información que se maneja según su objetivo, la calidad que debe tener la información quiere decir que se establezca cuando debe verificarse su integridad y cuando debe de verificarse su autenticidad tanto de la información como de los usuarios.

- **Análisis de Riesgos.**

  - Consiste en enumerar todo tipo de riesgos a los cuales esta expuesta la información y cuales son las características, los posibles atacantes entre personas, empresas, dependencias de inteligencia y las posibles amenazas.
  - Enumerar todo tipo de posible pérdida, desde pérdidas directas como dinero, clientes, tiempo, así como pérdidas de imagen y pérdidas de confianza.

  El riesgo se pude calcular mediante la fórmula:

  Riesgo = probabilidad x perdida

Por ejemplo:

El riesgo de perder un contrato por robo de información confidencial es igual a la probabilidad de que ocurra el robo, multiplicado por la pérdida total en pesos de no hacer el contrato.

El riesgo de fraude en transacciones financieras es igual a la probabilidad de que ocurra el fraude por la pérdida en pesos de que llegara a ocurrir ese fraude.

" Si la probabilidad es muy pequeña el riesgo es menor, pero si la probabilidad es casi uno, el riesgo puede ser casi igual a la pérdida total. Si por otro lado la pérdida es menor aunque la probabilidad de que ocurra el evento sea muy grande tenemos un riesgo menor."

Por ejemplo:

La pérdida de una transacción de 300 córdobas con una probabilidad muy grande de que ocurra al usar ***Criptografía*** débil, el riesgo llega a ser menor.

En el Análisis de Riesgo debe también incluirse los posibles ataques que puedan existir y sus posibles efectos.

- **Medidas de Seguridad.**

Esta parte la podemos plantear como la terminación de toda la estructura de seguridad de la información.
Una vez planteada una Política de Seguridad (decir cuanto vale la información), un Análisis de Riesgo (decir que tanto pierdo si le ocurre algo a mi información o que tanto se gana si se protege), debe de establecer las medidas para que cumpliendo con la Política de Seguridad, las pérdidas sean las menores posibles y que esto se transforme en ganancias ya sean materiales o de imagen.

Las posibles Medidas de Seguridad que se pueden establecer se pueden dividir según la siguiente tabla:

Tipos	*Protección Física*	*Medidas Técnicas*	*Medidas de Organización*
Preventivas	PF	PT	PO
Detectivas	DF	DT	DO
Correctivas	CF	CT	CO

Figura 2.

- **PF:** guardias a la entrada del edificio, control en el acceso de entrada, protección al hardware, respaldo de datos.

- **DF:** monitor de vigilancia, detector de metales, detector de movimiento.
- **CF:** respaldo de fuente de poder.
- **PT:** firewalls, ***Criptografía***, bitácora.
- **DT:** control de acceso lógico, sesión de autenticación.
- **CT:** programa antivirus.
- **PO:** cursos de actualización, organización de claves.
- **DO:** monitoreo de auditoria.
- **CO:** respaldos automáticos, plan de incidentes (sanciones).

Uno de los objetivos principales por el cual debemos de establecer una Política de Seguridad es, el de reducir al mínimo los riesgos posibles, implementando adecuadamente las diferentes medidas de seguridad.

Esto dicho, parece no revestir mayor importancia, la ***Criptografía*** se ha convertido en pieza clave de un debate que ha desbordado muchos foros restringidos, hasta configurarse como uno de los focos de mayor atención de la mayoría de los gobiernos del planeta.

En algunos países está directamente prohibido el uso de encriptación de mensajes (como Francia o China), en otros como Estados Unidos está fuertemente controlado, impidiéndose la exportación de programas encriptadores al considerarse por el Acta de Control de Exportación de Armas *(Arms Export Control Act)* como incluida en su lista, junto a misiles, bombas y armamento diversos.

Hay muchos países que, aunque en su territorio nacional permiten el uso de la ***Criptografía***, desean que estos programas incluyan una puerta trasera **(backdoor),** o procedimiento parecido para poder intervenir el mensaje cuando así lo consideren oportuno. Es el caso del famoso chip de depósito de claves o “Chip Clipper”, para encriptar conversaciones telefónicas (los dos teléfonos participantes en una conversación deben tenerlo).

Todo esto nos lleva directamente al enfrentamiento **“Privacidad en las Comunicaciones - Control Gubernamental”**, lo que en otros términos se denomina "El control del Gran Hermano" (aunque esta expresión se utiliza, también, para denominar a esa especie de *Ojo Vigilante*, que presuntamente nos acecha continuamente y cuyo orígen es indeterminado: Gobiernos, espías de distinto grado, fisgones o meros curiosos...), lo cual desemboca en la posible afectación de derechos fundamentales de las personas, como es el derecho a la Libertad de Expresión, que difícilmente se puede conseguir.

Cuando nos comunicamos con alguien no sabemos quién o quienes pueden realmente leer el mensaje; problema que se agrava en Internet, ya que los mensajes se pueden quedar en el ciberespacio por tiempo indefinido, sin tener nosotros siquiera conciencia de ello o de donde estará efectivamente copiada o almacenada nuestra comunicación.

La cuestión es conseguir que aunque nuestros mensajes puedan ser interceptados, resulten totalmente ininteligibles. Y esto se consigue con la ***Criptografía***.

La colisión de intereses que se produce es, por un lado el Derecho a la Intimidad y a la Privacidad, y por otro, el deseo de los Cuerpos de Seguridad de que no exista información a la que no puedan tener acceso.

En Estados Unidos, actualmente está abierto un gran debate y de gran interés; por un lado los defensores de la Privacidad, por otro, cifras como las que presenta el **FBI** (y eso que ellos no llevan a cabo la totalidad de las escuchas realizadas en los EE.UU). Entre 1985 y 1994, las escuchas ordenadas judicialmente formaron parte de las pruebas que concluyeron en 14.648 sentencias, supusieron casi 600 millones de dólares en multas y más de 1.500 millones de dólares en recuperaciones y embargos ordenados por los jueces. Esto se imposibilitaría con el uso de encriptación fuerte.
Lo primero que hay que establecer es qué aplicaciones necesitan seguridad y cuántos servicios se necesitan.
En segundo lugar hay que determinar cómo se van a proporcionar esos servicios, si van a ser transparentes al usuario o si se les va a dejar elegir el tipo de servicio.

También es necesario determinar en qué nivel se van a proporcionar, sí en el nivel de aplicación o en niveles inferiores. Y sobre todo, tanto si se utiliza ***Criptografía de Clave Secreta***, como si se utiliza ***Criptografía de Clave Pública*** es necesario diseñar un sistema de gestión de claves y definir una política que determine la forma en la que se debe operar.

Los beneficios de un sistema de seguridad bien elaborados son inmediatos, ya que la organización trabajará sobre una plataforma confiable, que se refleja en los siguientes puntos:

- Aumento de la productividad.
- Aumento de la motivación del personal.
- Compromiso con la misión de la compañía.
- Mejora de las relaciones laborales.
- Ayuda a formar equipos competentes.
- Mejora de los climas laborales para los RR.HH.

En resumen, de lo anteriormente escrito, podemos decir que no existe un sistema computarizado que garantice al 100% la **Seguridad de la Información** debido a la inmensa mayoría de formas diferentes con que se puede romper la seguridad de un Sistema Informático.

Sin embargo una buena planeación de la estrategia para dar seguridad a la información puede resultar desde la salvación de una empresa hasta la obtención de grandes ganancias directas en cuanto a dinero se refiere, mejorando la imagen y la seguridad de la empresa.

Capítulo 1

CONCEPTOS GENERALES

1.1. INTRODUCCIÒN.

En la actualidad, la falta de **Medidas de Seguridad** en las redes es un problema que está en crecimiento. Cada vez es mayor el número de atacantes y cada vez están más organizados, por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios en su labor de piratería.
La definición de un entorno seguro implica la necesidad de estudiar varios aspectos y de establecer una infraestructura que dé soporte a los **Servicios de Seguridad** que se quieren proporcionar.

El uso de técnicas criptográficas tiene como propósito prevenir algunas de las faltas de seguridad en un Sistema Computarizado. La Seguridad en general debe ser considerada como un aspecto de gran importancia en cualquier corporación que trabaje con sistema computarizado. El hecho que gran parte de actividades humanas sea cada vez más dependiente de los sistemas computarizados hace que la seguridad juegue un papel importante.

Hasta hace pocos años la ***Criptografía*** no resultaba interesante, para agencias de seguridad, gobiernos, grandes empresas y aún “delincuentes – informáticos”. Sin embargo, en poco tiempo, debido al rápido crecimiento de las comunicaciones electrónicas, ésta interesante ciencia se ha convertido en un tema sugerente que llama la atención del público en general. Destaca especialmente el cambio que ha sufrido, durante el final del siglo pasado, la orientación de la investigación en ***Criptografía***, ya que ha pasado del tema clásico del cifrado y su seguridad, hacia los más actuales campos de las **Firmas Digitales** y los **Protocolos Criptográficos.**

Dicha variación es una consecuencia inmediata del impacto de la informatización en la sociedad, que cada vez demanda más servicios telemáticos eficientes y seguros. Ante estas situaciones de peligro nacidas a raíz de los nuevos servicios, se hacen necesarias soluciones diferentes.

Esta es una de las razones que nos ha movido a elaborar el presente estudio, pero no la única, puesto que el conocimiento de esta materia es importante no solamente para estudiantes universitarios (a quienes se les imparte en diferentes especialidades de la Informática), sino también para profesionales que están en contacto con las diferentes fuentes documentales.

Por todo ello la finalidad de este manual es recorrer los aspectos más destacables de cada una de las facetas en las que se ve envuelta de la ***Criptografía Moderna***.

Se retienen aquí tanto los fundamentos de la base teórica, como las descripciones y el análisis de los sistemas de **Clave Secreta** y **de Clave Pública** más conocidos, difundidos y utilizados actualmente.

La estructura de los diferentes aspectos que abordaremos en las próximas páginas lo trataremos de hacer posiblemente de la manera más sencilla; ya que como hemos apuntado con anterioridad, nuestra voluntad es confeccionar un manual introductorio accesible para todos los que se interesen por esta disciplina.

En primer lugar, estableceremos una serie de nociones generales acerca de la ***Criptografía***: definiciones de los términos mas usados y su finalidad. En segundo lugar, analizaremos los principales protocolos criptográficos o algorítmos que se ven envueltos en la ***Criptografía Moderna*** empleados en la elaboración de Sistemas cifrados de mayor relevancia, que permiten resolver problemas en el mundo de las telecomunicaciones (Identificación de usuarios para el Control de Acceso).

De igual manera, hemos incluido variadas ilustraciones o gráficos en el presente manual, sobre aspectos tratados en este estudio, con el fin de facilitar la comprensión de los mismos.

Por último, hemos creído oportuno incluir una aplicación (software) llamado **"CriptoSistemas Modernos de Clave Privada y Pública"** proporcionado por el Dr. Jorge Ramió Aguirre, profesional de alto rango en el tema y la materia, catedrático y profesor de la U.P.M (Universidad Politécnica de Madrid), que demuestre cómo es el funcionamiento de un **CriptoSistema Moderno** (utilizando algoritmos de encriptación moderna), concluyendo así el propósito de nuestro Proyecto en general.

1.2. BREVE HISTORIA DE LA CRIPTOGRAFÌA.

Desde que el hombre dispuso de la escritura como vehículo de comunicación, mostró un empeño especial en impedir la lectura de información particular.

Los sistemas más sencillos empleados en un principio para enviar mensajes privados consistieron en receptáculos cerrados en los que se guardaba la información; pero bastaba con capturar al portador para obtenerla, lo que hizo necesario encubrir de alguna forma el contenido del mensaje, para que su localización no conllevara su interpretación.

Los usos más primitivos de la ***Criptografía*** se encuentran documentados desde la época de Julio César. Estos mecanismos de cifrado se basaban en técnicas de transposición de caracteres y fundamentan su eficacia en el secreto del algorítmo empleado para el cifrado. Algorítmos de este tipo son sólo de interés histórico.

Entendida en sentido amplio, la ***Criptografía*** se usa desde la más remota antigüedad, pues indios, chinos, persas, asirios, babilonios y egipcios poseían ya signos convencionales, equivalentes a las letras de sus alfabetos, con los que comunicaban órdenes secretas a sus emisarios, especialmente en tiempo de guerra, y los que daban en ocasiones, además de este valor práctico, unos atributos mágicos y religiosos.

A lo largo del tiempo de la historia se han empleado diferentes **Sistemas Criptográficos** para lograr la ocultación de la información, por ejemplo:

- **Criptografía Clásica:** la cual se basa en dos procedimientos básicamente:

  - ***Transposición,*** inventado por los griegos, y que consiste en barajar los símbolos del mensaje original colocándolos en un orden distinto, de manera que el criptograma contenga los mismos elementos de texto claro pero colocados de tal forma que resulten incomprensibles. El receptor, con conocimiento de la Transposición realizada, recoloca los símbolos desordenados del criptograma en su posición original.

  - ***Sustitución,*** inventados por los romanos al final de la República, que consiste en establecer una correspondencia entre las letras del alfabeto en el que está escrito el mensaje original y los elementos de otro conjunto que puede ser el mismo o distinto alfabeto. De esta forma, cada letra de texto claro se sustituye por su símbolo correspondiente en la elaboración del criptograma. El receptor al que va dirigido el mensaje, que deberá conocer la correspondencia establecida, sustituye cada símbolo del criptograma por el símbolo correspondiente del alfabeto original, recuperando la información original.

- **Criptografía Moderna:**

  - ***Criptografía Asimétrica.***

- ***Criptografía Simétrica.***
  Las cuales hemos preferido analizar de manera independiente en el transcurso del manual por ser principalmente la base de estudio de nuestro proyecto; existiendo sí una serie de métodos que dan un fuerte apoyo y soporte a la ***Criptografía Moderna.***

Para dar ejemplo de cómo la ***Criptografía*** a existido a través del tiempo, veamos como lo hizo en épocas antiguas.
El método que empleo Damarato en el siglo VI A.C, para informar a los lacedemonios del proyecto de Jerjes de invadir a Grecia. Según relata Herodoto, este rey espartano escribió un mensaje en una tablilla y la recubrió después de cera.

Existe un sector muy amplio en el que se superponen el estudio de las lenguas y el de la ***Criptografía***:

- Todo el campo que se requiere al de las lenguas muertas.
- Gracias a la piedra Rosetta, se descifró en 1822 los jeroglíficos del egipcio arcaico.
- En la primera mitad del siglo pasado lograron descifrar los caracteres cuneiformes con que se escribían las lenguas de los antiguos Persas, de los Asirios y de los Babilonios.

1.3. NOCIONES GENERALES.

1.3.1. INTRODUCCIÓN.

Para establecer una comunicación de datos entre dos entidades (personas, equipos informáticos, etc.) hacen falta al menos tres elementos básicos: **El Emisor del mensaje** *(la fuente)*, **El Receptor del mismo** *(el destino)* y un **Soporte Físico** por el cual se transfieran los datos *(el medio)*.

En una comunicación normal los datos se envían a través del medio tal como son, sin sufrir modificaciones de ningún tipo, de tal forma que el mensaje que representan puede ser interceptado y leído por cualquier otra entidad que acceda a él durante su viaje por el medio.

Pero hay ocasiones en las que nos interesa que dicho mensaje sólamente pueda ser interpretado correctamente por el Emisor del mismo y por el Receptor al que va dirigido. En estas ocasiones es necesario implementar algún mecanismo de protección de la información sensible, tal que el mensaje viaje seguro desde la fuente al destino, siendo imposible la interceptación por terceros del mensaje, o si ésta se produjese, el mensaje capturado sea incomprensible para quien tenga acceso al mismo.

Una de las formas de conseguir esto es enviar el mensaje en claro, tal como lo ha redactado el emisor, y protejerlo en el camino mediante sistemas de fuerza que lo defiendan durante el camino, (como es el caso de la protección de mensajes mediante un personal o guardias de seguridad).

Otro método posible es el enviar el mensaje por un camino con tanto tráfico de información que resulte muy difícil a las terceras personas detectar que se trata de información confidencial, (como es el caso de enviar una carta por el sistema estándar de correo).

Desafortunadamente estos métodos de protección de mensajes, al igual que otros análogos, han demostrado su inefectividad a lo largo de los tiempos, por lo que hubo que buscar otro tipo de mecanismos para proteger la información sensible en su camino entre emisor y receptor.
Es entonces cuando la ***Criptografía*** aparece con objeto de proporcionar comunicaciones seguras sobre canales inseguros

La ***Criptografía*** ha demostrado con el tiempo ser una de las mejores técnicas para resolver este problema. Tanto es así que actualmente, en la época de las computadoras y la información, es el mecanismo más usado en los procesos de protección de datos, como las transacciones bancarias por Internet, el correo electrónico cifrado, etc.
Esto es así porque es tal vez el único medio accesible y fácil de implementar para lograr un acceso controlado a la información en un medio, Internet, que por su propia naturaleza es abierto y de acceso libre a la información.

A diferencia de la ***Criptografía Clásica*** la ***Criptografía Moderna*** (que aparece a partir de los años 50's y tiene como base la complejidad computacional), y contempla los siguientes factores:

- Velocidad de Cálculo: con la aparición de las computadoras se dispuso de una potencia de cálculo muy superior a la de los métodos clásicos.

- Avance de las Matemáticas: que permitieron encontrar y definir con claridad sistemas criptográficos estables y seguros.

- Necesidades de Seguridad: surgieron muchas actividades nuevas que precisaban la ocultación de datos, con lo que la rama "**Criptología"** experimentó un fuerte avance.

Los algorítmos modernos usan una clave para controlar el cifrado y descifrado de los mensajes. Generalmente, el algorítmo de cifrado es públicamente conocido y sometido a escrutinio por parte de expertos y usuarios. Se acepta, por tanto, la denominada **Hipótesis de Kerckhoffs**, que establece que ***"La seguridad del cifrado debe residir, exclusivamente, en el secreto de la clave y no en el del mecanismo de cifrado".***

Tanta ha sido la importancia de los sistemas criptográficos a través del tiempo, que por ejemplo, en la Segunda Guerra Mundial la famosa máquina alemana **"ENIGMA",** trajo en jaque durante mucho tiempo al ejercito aliado, al permitir a los nazis el envío de información cifrada a sus tropas.

" Máquina ENIGMA "

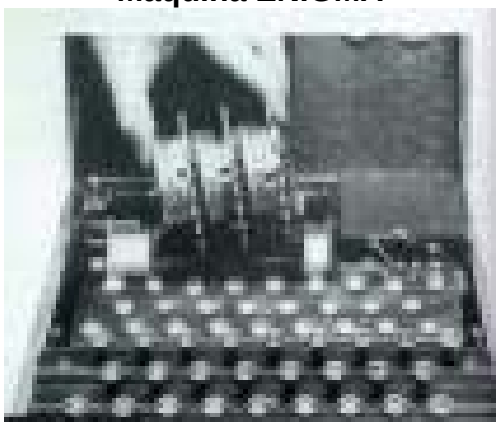


Figura 1.1.1.

En la actualidad los sistemas de cifrado están financiados en su mayoría por los gobiernos y sus militares, constituyendo el resultado de las investigaciones materia reservada.

1.3.2. CRIPTOGRAFÍA.

Él término ***Criptografía*** proviene de dos vocablos griegos: ***KRYPTOS,*** que significa esconder o escondido y ***GRÁPHEIN,*** que significa escribir o escritura. Más tarde se añade el sufijo **–ía** para conferirle el carácter de conocimiento o tratado.

Según esta definición de carácter etimológico, entendemos por ***Criptografía*** la técnica de transformar un mensaje, denominado **Texto en Claro** (también llamado, según una no muy adecuada traducción, **texto plano**), en otro que sólo puedan entender las personas autorizadas a ello, que llamaremos **Criptograma** o **Texto Cifrado**.
El método o sistema empleado para encriptar el texto en claro se denomina **Algorítmo o Protocolo de Encriptación**.

La ***Criptografía*** es una rama de las Matemáticas, que se complementa con el **Criptoanálisis**, que es la técnica de descifrar textos cifrados por criptógrafos, sin tener autorización para ellos, es decir, realizar una especie de ***Criptografía Inversa***. Ambas técnicas forman la ciencia llamada **Criptología,** el cual se encarga del estudio y práctica de los sistemas de cifrado destinados a ocultar el contenido de mensajes enviados entre dos partes: **Emisor** y **Receptor;** los cuales quieren intercambiar mensajes. El posible enemigo que quiere interferir de algún modo en la comunicación se denomina **Intruso**. Este intruso puede ser pasivo, si sólo escucha la comunicación, o activo si trata de alterar los mensajes

Pero aún se puede precisar mas este concepto, y así, esta disciplina, es entendida como el arte de escribir en un lenguaje convenido mediante el uso de **Claves o Cifras**, es decir, la ***Criptografía*** enseña a diseñar cifrarios (expresión sinónima de código secreto o escritura secreta).

La base de la ***Criptografía*** suele ser la aplicación de problemas matemáticos de difícil solución a aplicaciones específicas, denominándose **CriptoSistemas** o **Sistema de Cifrado** a los fundamentos y procedimientos de operación involucrados en dicha aplicación

Otro concepto de suma importancia que debemos de conocer, en el ámbito del tema es el de **Cifrar** o **Encriptar,** que simplemente es cuando se manda un mensaje con información confidencial. La labor de transformar un texto cifrado en el mensaje original, si se conoce como la clave, se denomina **Descifrar** o **Decodificar**, mientras que si se ignora el código secreto es más adecuado llamarla **Perlustrar** o **Descriptar**.

El texto en claro se representa como **M** (por *message*) o también por **P** (de *plain text*). **M** es simplemente un dato binario. El texto cifrado se designa por **C** (de *ciphertext*). No existe relación directa entre los tamaños de ambos mensajes. Unas veces su tamaño coincide. Otras, el del texto cifrado es mayor que el del texto en claro. Puede ocurrir, incluso, que texto cifrado sea de menor tamaño. Esto sucede cuando, además de las técnicas de cifrado, se emplean técnicas de compresión.

La función de cifrado, **E,** opera sobre **M** para producir **C**. En notación matemática:

- E(M) = C

Inversamente, la función de descifrado, **D**, se aplica a **C** para producir **M**:

- **D(C) = M**

En todo caso, debe cumplirse la siguiente igualdad:

- D(E(M)) = M

En atención a lo expuesto, la “Criptología”, “Perlustración” o ”Descriptación”, términos equivalentes, es la actividad que tiene por objeto el descifrado de criptogramas o documentos configurados con elementos secretos desconociendo la clave.

Sin duda, esta matización es importante a la hora de emplear los diversos términos definitorios. Por consiguiente, en la terminología más exacta se hace una distinción entre descifrar y descriptar:
Descifrar: el destinatario legitimo conoce la clave o norma secreta mediante la cual el texto original se transforma en el texto codificado y descifra el criptograma;
Descriptar: el espía intenta descubrir el texto cifrado.

El problema inmediato que se plantéa en cualquier sistema complejo, es recordar el nuevo orden que hemos establecido para obtener el mensaje camuflado, problema tanto más difícil de resolver cuanto más complicado haya sido el sistema elegido.

El uso de algunas palabras o serie determinada como base de un sistema de cifrado posee la ventaja de que, si el sistema es complejo, tan sólo será fácil obtener el texto en claro a quién sepa dicha palabra, además de ser fácil de recordar.
Esta palabra o serie base del mecanismo de cifrado se denomina **Clave de Cifrado**, y el número de símbolos que la forman se llama **Longitud de la Clave**.

Indudablemente, cuanto más complicado sea el mecanismo de cifrado y cuanto más larga sea la clave, más difícil será romper el sistema y obtener el mensaje original para un extraño. Pero más complicado será también para el destinatario del mensaje cifrado realizar las operaciones de descifrado y obtener el mensaje original, por lo que se crea el dilema “SEGURIDAD / TIEMPO”.

Las claves de encriptación van a ser la base fundamental de los modernos sistemas criptográficos, basados en operaciones matemáticas generalmente muy complejas.

Pero la ***Criptografía*** no es en sí seguridad; simplemente es la herramienta utilizada por mecanismos más complejos para proporcionar no sólo confidencialidad, sino también otros servicios de seguridad, ya que, la

confidencialidad (en el contexto de Internet) es, a menudo, un factor secundario. Generalmente estaremos más interesados en el mantenimiento de la integridad de los mensajes y en los mecanismos de autenticación que, implícitamente, proporciona la ***Criptografía***.

Podemos deducir entonces que para **Criptoanalizar** un documento cifrado o criptograma, hay que saber la clave del significado de los signos, es decir el sistema o el código seguido, ya que si no será muy difícil interpretar su contenido.
La formación de la clave no es difícil, puesto que no se sujeta a reglas fijas y solo depende de la mayor o menor pericia en la combinación de los signos y saber el protocolo empleado.

En cuanto a la nomeclatura, aparte del nombre de ***Criptografía***, este sistema de escribir ha recibido a lo largo de la historia otros nombres como: Poligrafía, Estenografía, Pasigrafía y Esteganografía. A este tipo de escritura se le ha denominado "cifrada", "diplomática" o "en clave", por ser usada frecuentemente por los gobiernos de las naciones y sus representantes diplomáticos.

Algunos Criptógrafos dividen el relieve de la ***Criptografía*** en dos partes:

- **Estrategia:** Consiste en garantizar el secreto de los mensajes cifrados por un largo período de tiempo. En este tipo de ***Criptografía*** se tiene que ser muy prudente.
- **Táctica:** Se conforma con una duración menor, la necesaria para llevar a buen termino una acción determinada. Se pueden tomar las cosas más a la ligera.

Así con lo mencionado hasta ahora, podemos afirmar que la finalidad de la ***Criptografía*** es, sin duda, ocultar a terceras personas, el contenido de textos que no les han sido destinados o que por su naturaleza e importancia solo los deben conocer los interesados (Emisor – Receptor).

1.4. PROBLEMAS DE SEGURIDAD QUE RESUELVE LA CRIPTOGRAFÌA.

1.4.1. PRIVACIDAD.

Se refiere a que la información solamente puede ser leída por personas autorizadas.

Ej. Si la comunicación se establece por Teléfono y alguien intercepta la comunicación o escucha la conversación por otra línea podemos afirmar que no existe Privacidad.

Ej: Si mandamos una carta y por alguna razón alguien rompe el sobre para leerla, podemos decir que se ha violado la Privacidad.

En la comunicación por Internet es muy difícil estar seguro que la comunicación es privada, ya que no se tiene control de la línea de comunicación. Por lo tanto si ciframos (escondemos la información), cualquier intersección no autorizada no podrá entender la información confidencial. Esto es posible si se usan técnicas criptográficas, en particular la Privacidad se logra si se cifra el mensaje con un Método Simétrico.

1.4.2. INTEGRIDAD.

Se refiere a que la información no pueda ser alterada en el transcurso de ser enviada.
Ej. Cuando compramos un boleto de avión es muy prudente verificar que los datos son los correctos antes de terminar la operación, en un proceso común esto se puede realizar al mismo tiempo de la compra, por Internet como la compra se puede hacer desde dos ciudades muy distantes y la información tiene necesariamente que viajar por una línea de transmisión de la cual no se tiene control.

Es muy importante estar seguros que la información transmitida no ha sido modificada (en tal caso se dice que hay Integridad). Esto también se puede solucionar con Técnicas criptográficas particularmente con procesos Simétricos o Asimétricos. La Integridad es muy importante porque en un cambio de información puede causar graves problemas.

1.4.3. AUTENTICIDAD.

Se refiere a que se pueda confirmar que el mensaje recibido haya sido mandado por quien dice haberlo enviado o que el mensaje recibido es el que se esperaba.
Ej. Las técnicas necesaria para poder verificar la autenticidad tanto de personas como de mensajes usando quizás la mas conocidas aplicación de la ***Criptografía Asimétrica*** que es la **Firma Digital** y de algún modo reemplaza a la Firma Autógrafa que se usa comúnmente.

Por Internet es muy fácil engañar a una persona con quien se tiene comunicación respecto a la identidad; resolver este problema es por lo tanto muy importante para efectuar comunicación confiable.

1.4.4. NO RECHAZO.

Se refiere a que no se puede negar la autoría de un mensaje enviado. Cuando se diseña un sistema de seguridad una gran cantidad de problemas pueden ser evitados si se ponen en función de comprobar autenticidad, de garantizar Privacidad de asegurar integridad y evitar el no rechazo

1.5. COMPROMISO ENTRE CRIPTOSISTEMA Y CRIPTOANALISIS.

Tanto los **CriptoSistemas** como el **Criptoanálisis** están ligados fuertemente, ya que pueden existir sistemas idealmente seguros, capaces de resistir cualquier ataque. También veremos que estos sistemas en la práctica carecen de interés, lo cual nos lleva a tener que adoptar un compromiso entre el costo del sistema, tanto computacional como de almacenamiento, e incluso económico frente a su capacidad de resistencia a diferentes ataques criptográficos.

La información posee un tiempo de vida, y pierde su valor transcurrido el mismo.
Ej: Los datos sobre la estrategia de inversiones a largo plazo de una gran empresa, tienen un mayor período de validez que la exclusiva periodística de una sentencia judicial que se va a hacer pública al día siguiente.

El ejemplo anterior hace énfasis en el *"Que el tiempo en que un sistema se puede tardar en comprometer su seguridad es mayor que el tiempo de vida de la propia información que este alberga"*.
Esto no suele ser fácil, sobre todo porque no tardará lo mismo un oponente que disponga de una única computadora de capacidad modesta, que otro que emplee una red de supercomputadores.

Por eso también debe de tenerse en cuenta; si la información que queremos proteger vale mas que el esfuerzo de Criptoanálisis que va a necesitar, porque entonces puede que no este segura. La seguridad de los CriptoSistemas se suele medir en términos del número de computadoras y del tiempo necesario para romperlos, y a veces simplemente en función del dinero necesario para llevar a cabo esta tarea con garantías de éxito.

De cualquier forma, hoy por hoy existen sistemas que son muy poco costosos o 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 será insuficiente para romperlos.

Además no es conveniente confiar extremadamente en el algoritmo de cifrado, puesto que en el proceso de protección de la información existen otros puntos débiles que deben ser tratados con mucho cuidado.
Ej: No tiene sentido emplear algoritmos con niveles de seguridad extremadamente elevados si luego escogemos contraseñas (passwords) ridículamente fáciles de adivinar. Una práctica muy extendida de lo antes dicho, por desgracia, es la de escoger palabras claves que contengan fechas, nombres de familiares, nombres de personajes o lugares de ficción, etc.
Son las primeras que un atacante avispado probará. Tampoco es una práctica recomendable anotarlas o decírselas a nadie, puesto que si la clave cae en malas manos, todo nuestro sistema queda comprometido, por buenos que sean los algorítmos empleados.

1.6. FORTALEZA DE UN ALGORÌTMO CRIPTOGRÀFICO.

En la actualidad prácticamente todas las aplicaciones criptográficas emplean computadoras en sus cálculos y las computadoras convencionales están diseñadas para ejecutar algorítmos por tanto, definiremos como **Algorítmo** a una secuencia finita y ordenada de instrucciones elementales que, dados los valores de entrada de un problema, en algún momento finalizan y devuelven la solución.

Un buen sistema criptográfico debe ser diseñado de modo que su rotura sea tan difícil como sea posible. En la práctica, un buen sistema es aquel que no puede

ser roto en la práctica (aunque teóricamente pueda serlo). Sin embargo, esto no es, a menudo, fácil de demostrar.

Un **Algorítmo Criptográfico** es considerado seguro sí:

- No existen puertas traseras. Es decir, no hay ningún método para recuperar un mensaje en claro a partir del mensaje cifrado sin utilizar búsquedas exhaustivas de la clave.

- El número de claves posibles es lo suficientemente grande como para que la búsqueda exhaustiva no sea práctica.

Teóricamente, cualquier algorítmo basado en el uso de una clave puede ser roto probando todas las claves posibles. Este método es conocido como Ataque basado en la Fuerza Bruta.

Si este es el único método posible (se asume la inexistencia de puertas traseras), la potencia de computación necesaria crece exponencialmente con la longitud de la clave. Centrándonos en algorítmos de clave simétrica.

Por ejemplo:

Una clave de 32 bits de longitud requiere 2^{32} operaciones (aproximadamente 10^9). En media, 2^{31} operaciones deberán bastar para encontrar la clave correcta. Este cálculo puede ser abordado por cualquier particular en su ordenador personal. Si la longitud fuese de 40 bits (como, por ejemplo, la versión exportable de EE.UU. y Canadá de RC4) se necesitarían 2^{40} operaciones. Esta potencia de cálculo está al alcance de una universidad de tamaño medio o de, incluso, algunas pequeñas empresas. Un sistema con una clave de 56 bits (como la de **DES**) requiere un esfuerzo substancial, pero abordable con un equipamiento específico. Este equipamiento está al alcance de organizaciones criminales, grandes compañías y gobiernos.

Existe una regla empírica, conocida como la **Ley de Moore**, que establece que ***"La potencia de computación disponible para una inversión monetaria fija se dobla, aproximadamente, cada año y medio"***.
Por tanto, para mantener los actuales parámetros de protección, habría que añadir un bit a la clave cada dieciséis meses.
Los algoritmos de encriptación pueden tener dos tipos de secretos: el **Secreto Teórico** o **Incondicional** y el **Secreto Práctico** o **Computacional.**

- *Un CriptoSistema tiene secreto Teórico*, si es seguro contra cualquier enemigo que tenga recursos y tiempo ilimitados.
- *Un CriptoSistema tiene secreto Práctico*, si es seguro contra aquellos enemigos que tengan menos de una cantidad de tiempo y/o recursos.

Sistemas con claves de 64 bits son invulnerables en la actualidad, pero serán atacables en pocos años.

Finalmente, sistemas con claves de 128 bits permanecerán resistentes a ataques basados en la fuerza bruta en un futuro previsible. Es de destacar, además, que el coste derivado de usar claves seguras (de 128 o más bits de longitud) no es significativamente mucho mayor que en el que incurren cifrados con claves "débiles".

En el caso de los algorítmos de clave asimétrica, las claves son mucho más largas. El problema no es ahora adivinar la clave, sino deducir la clave privada a partir de la clave pública. En el caso del algorítmo **RSA**, esto es equivalente a factorizar un entero muy grande con dos factores primos también grandes. Para dar una idea de la complejidad implicada, una clave de 256 bits puede ser factorizada fácilmente por particulares en cualquier PC. Claves de 384 bits pueden ser deducidas por universidades o compañías de tamaño medio. 512 pueden ser rotas por gobiernos, y claves de 768, aunque son seguras en la actualidad, no lo son en un futuro próximo. Claves de 1024 bits pueden ser consideradas seguras en tanto que no se avance en las técnicas de factorización de enteros. De mantenerse dicha hipótesis, claves de 2048 se consideran seguras durante las próximas décadas.

1.7. QUIÉN ES QUIÉN EN EL MUNDO DE LA CRIPTOGRAFÍA.

En el ámbito de la ***Criptografía*** existen grandes entidades, agencias o empresas que regulan el manejo, uso y aplicación de la misma, entre las cuales están:

1.7.1. NSA (*NATIONAL SECURITY AGENCY*).

La **NSA** o Agencia de Seguridad Nacional de los EE.UU fue creada en 1952 por el presidente Truman, su existencia era incluso negada hasta hace comparativamente poco tiempo; su presupuesto o el tamaño de su plantilla siguen estando clasificados.

La **NSA** depende del Departamento (Ministerio) de Defensa y tenía como objetivo interceptar e interpretar cualquier comunicación que fuera de, presumible, interés para la seguridad de los EE.UU.
Para ello, se dedica a la investigación criptológica en dos vertientes: por un lado diseña algoritmos de cifrado seguros para proteger las comunicaciones de los EE.UU., y por otro diseña técnicas criptoanalíticas capaces de "reventar" cualquier comunicación de interés.

Por su propia naturaleza, la **NSA** se comporta como un *lobby* de presión tendente a evitar cualquier uso no controlado de ***Criptografía*** "fuerte" o, directamente, a prohibirla.

1.7.2. NCSC (*NATIONAL COMPUTER SECURITY CENTER*).

El **NCSC,** es una rama de la **NSA** responsable del programa de computación segura del gobierno de EE.UU. A tal efecto, el **NCSC** evalúa productos de seguridad *software* y *hardware* comerciales, efectúa trabajos de investigación y proporciona asesoría técnica y formación.

NCSC es famoso, sobre todo, por su serie de libros dedicados a la seguridad informática. Esta serie se conoce popularmente como los libros "arco iris" debido a que cada uno posee una portada de diferente color. El más famoso de la serie el conocido como "Libro Naranja", oficialmente *Department of Defense Trusted Computer System Evaluation Criteria*.
Este documento trata de definir requisitos de seguridad, de modo que sea sencillo para fabricantes medir de forma objetiva la seguridad de sus sistemas. Se centra en la seguridad y apenas trata de ***Criptografía.***

Otro libro significado el "Libro Rojo", oficialmente conocido como *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, que como su nombre indica, interpreta los requisitos del Libro Naranja para redes.

1.7.3. NIST (*NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY*).

El **NIST** pertenece al Departamento (Ministerio) de Comercio de EE.UU. Anteriormente, se conocía como *National Bureau of Standards* (NBS). El **NIST** promueve la interoperabilidad y la definición de estándares abiertos para conseguir el desarrollo de la industria informática. Para ello, publica una serie de estándares y recomendaciones con objeto de que sean adoptadas por la industria.

Los estándares oficiales se publican como FIPS (*Federal Information Processing Standards*.

NIST ha desarrollado **DES**, **DSS** o **SHS**. Estos estándares (y los correspondiente algorítmos) fueron desarrollados con un grado desconocido de colaboración por

parte de la **NSA**, de modo que siempre ha habido cierto recelo acerca de la "calidad" de los productos desarrollados por **NIST**.

1.7.4. RSA DATA SECURITY, INC.

Esta compañía fue fundada en 1982 por los creadores del algorítmo **RSA**, la base de su negocio es la comercialización, desarrollo y licenciamiento de la patente de **RSA**. También posee la licencia de **RC2** y **RC4.** Vende bibliotecas de cifrado y desarrolla los estándares criptográficos conocidos como **PKCS**. Posee un laboratorio de investigación criptográfico, los **RSA Laboratories**.

1.7.5. VERISING INC.

Es una compañía estadounidense fundada en 1995 como una rama de **RSA Data Security, Inc.** Es el principal suministrador mundial de certificados (es una de las autoridades de certificación predeterminadas en cualquier navegador). También desarrolla soluciones integradas en el campo del comercio electrónico.

Capítulo 2

HERRAMIENTAS DE SEGURIDAD

Dentro de el amplio marco, en el que se ve evuelta la ***Criptografía Moderna***, cabe señalar algunas **Herramientas de Seguridad** que nos van a dar soporte y un mejor entendimiento del tema como tal, entre las cuales están: Los **Servicios de Seguridad** y **Mecanismos de Seguridad**. Además debemos conocer algunos de los tipos de ataques más comunes que pueden afectar a cualquier sistema informático si no se cuenta con la debida seguridad.

2.1. SERVICIOS DE SEGURIDAD.

Los **Servicios de Seguridad** realzan la seguridad de los sistemas de procesamiento de datos y las transferencias de la información de una organización.
Dentro del modelo de referencia OSI, se define una arquitectura de seguridad, de acuerdo con esta arquitectura, se pueden proteger las comunicaciones de los usuarios a través de una red. Estos se clasifican de la siguiente manera:

2.1.1. TRANSPORTE DE DATOS.

2.1.1.1. Confidencialidad:

Este servicio, evita que se revelen deliberada o accidentalmente, los datos de una comunicación. La información es accesible solo para las personas autorizadas. En algunos casos no debe ser posible realizar un análisis del trafico.

2.1.1.2. Integridad:

Este servicio verifica que los datos de una comunicación no se alteren, esto es, que los datos recibidos por el receptor coincidan por los enviados por el emisor.

Garantiza que los datos no hayan sido modificado, alterados su orden o retrasados.

2.1.1.3. Autenticación:

Este servicio verifica la fuente de los datos. La Autenticación puede ser sólo de la entidad orígen, de la entidad destino o de ambas a la vez. Garantiza que la identidad del interlocutor sea la correcta.

2.1.1.4. No Repudio (Irrenunciabilidad):

El transmisor o receptor no pueden negar haber enviado o recibido.
Este servicio proporciona la prueba, ante una tercera parte, de que cada una de las entidades han participado, efectivamente, en la comunicación. Esta puede ser de dos tipos:
- Con prueba de orígen o emisor: el destinatario tiene garantía de quién es el emisor concreto de los datos.
- Con prueba de entrega o receptor: el emisor tiene prueba de que los datos de la comunicación han llegado íntegramente al destinatario correcto en un instante dado.

2.1.2. ACCESO A RECURSOS.

2.1.2.1. Control de Acceso:

Este servicio verifica que los recursos son utilizados por quién tiene derecho a hacerlo. Debe ser controlado el acceso de información a través de la red.

2.1.2.2. Disponibilidad:

Un sistema debe estar disponible por las partes autorizadas cuando sea necesario (prevención y recuperación frente a arranques físicos del sistema).

Cualquier sistema de transferencia segura basado en ***Criptografía Moderna*** debería abarcar por lo menos los primeros cuatro aspectos, pero no suelen hacerlo en su totalidad. Así, los sistemas de clave simétrica o privada ofrecen Confidencialidad, pero ninguno de los otros factores, en cambio los sistemas de clave asimétrica o pública ofrecen Autenticidad, Integridad, Confidencialidad en el envío (pero no en las fases posteriores) y No Repudio si van asociados a una Firma Digital.

Otro aspecto que debemos de tomar en cuenta en lo que se refiere a Seguridad en las Comunicaciones, aunque de cierta forma se salga del campo de la ***Criptografía***, es el de los **Servicios de Autorización**, que proporciona al usuario acceso solamente a los recursos a los que está autorizado.

Esta funcionalidad se suele implementar mediante servidores especiales al efecto, que administran bases de datos con los documentos a los que tiene permitido el acceso cada uno de los usuarios del sistema.

2.2 MECANISMOS DE SEGURIDAD.

2.2.1. CIFRADO.

El Cifrado es utilizado para ofrecer servicio de Confidencialidad de los datos. Normalmente lo hace mediante cifrados con claves simétricas.
Se pueden utilizar sistemas de:
Clave secreta: más rápido, importante cuando hay que aplicarlo a gran cantidad de datos.
Clave pública: más sencillo el problema de gestión de claves.
La codificación puede ser, entre extremo finales, en los nodos de cada enlace (en los protocolos del nivel 2 o 3).

2.2.1.1. Tipos de Cifrado.

Existen dos tipos de cifrados:

1. **Cifradores de Bloques**: que cifran bloques de tamaño fijo, normalmente de 64 bits.
2. **Cifradores de flujo**: los cuales se cifran sobre flujos continuos de bits.

2.2.1.2. Esquema de Cifrado.

Este esquema de cifrado se usa principalmente en cifrar claves de sistemas simétricos (claves de 128 bits aproximadamente).

1. Se toma el mensaje **M** (por ejemplo una clave simétrica de 128 bits), como en la práctica actual es recomendable usar arreglos de longitud de 1024 bits, los complementa esos 128 bits con una serie de técnicas para obtener un arreglo de 1024 bits, que la computadora entiende como un numero entero *m*, este proceso se llama ***codificación***.

2. Se le aplica la formula de cifrado de **RSA** al entero ***m.***

3. Se envía el número entero ***c.***

4. Al recibir este número se aplica la técnica de descifrado al entero ***c*** para obtener el entero ***m.***

5. Se descodifica *m* para obtener el mensaje **M.**

2.2.2. INTERCAMBIO DE AUTENTICACIÓN.

Es un protocolo que garantiza la Autenticidad de los dos interlocutores de una comunicación.

2.2.2.1. Introducción.

La Autenticación nos asegura que nuestro interlocutor es auténtico (no es un impostor).
Esta ofrece en conjunto con la Integridad, que la información recibida no ha sido alterada. Aunque los datos vengan de la persona correcta debe de verificar que:
- Su contenido no ha sido modificado.
- No han sido retenidos más tiempos del habitual.
- La secuencia de los mensajes no ha sido alterada.
- No se ha producido una repetición intencionada de un mensaje.

La Autenticación da soportes a otros mecanismos como el Control de Acceso. Para saber si alguien puede acceder a un recurso, primero hay que tener la seguridad de que no es un impostor.
Puede utilizar el cifrado para aportar Confidencialidad a toda o parte de la información.

2.2.2.2. Tipos de Autenticación.

- **Autenticación basadas en CriptoSistemas de Clave Privada.**

La ***Autenticidad*** se refiere a que la clave solo la conoce el otro interlocutor.

Antes de codificar:
- *Contenido no modificado:* incluir información de detección de errores.
- *No retención:* incluir marca de tiempo de salida.
- *Alteración de secuencia:* incluir número de secuencia.
- *Unicidad:* añadir números aleatorios(únicos).

Intercambio de Autenticación:
- *Validación de identificación:* procesos que permiten a los dos interlocutores que van a establecer una sesión autenticarse entre sí.
- Posiblemente se intercambie una clave K para usar en la sesión.

Hay que resolver el problema de la gestión de las claves.
- Protocolos de intercambio de claves.
- Centro de distribución de claves.

- **Autenticación basadas en CriptoSistemas de Clave Pública.**

- Se codifica con la clave privada.
- Todo el mundo lo puede decodificar.
- Solo el proveedor de la clave privada lo ha podido codificar.

Ventajas:

- Mejora en la distribución de claves.
    - Claves públicas y por lo tanto no secretas.
    - Una sola clave.
- Lo pueden codificar varios, útil para aplicaciones de multidifusión.

Inconvenientes:

- La codificación se realiza con clave públicas, es más lenta que con la clave secreta(100..1000 veces).

Se puede utilizar en una primera fase de validación de identificación en la que se intercambian una clave secreta para la sesión K.

- **Autenticación basadas en Métodos No Criptográficos.**

Son aplicaciones en las que no se necesita Confidencialidad y sí Autenticación. Se buscan métodos más rápidos.

Métodos:
- El transmisor y receptor comparten una clave.
- Se calcula información de autenticación MAC, función de la clave y el mensaje.
- Se concatena esta información al mensaje en claro y MAC.
- El receptor realiza los mismos cálculos y compara MAC obtenido con el recibido.
- Se puede añadir previamente: números de secuencias, detección de errores y marca de tiempo.

Algunos algoritmos de este tipo de Autenticación son: **MAC, HMAC - MD5, HMAC - SHA, HMAC - RIPEMD.**

2.2.2.3. Funciones HASH (De Resumen).

Genera un resumen (huella digital), **H** de tamaño fijo a partir de un mensaje de entradas.
Este resumen se encripta con sistemas de clave secreta o pública y se envía.
Al encriptar solo el resumen, se ahorra en tiempo de computo.
Una herramienta fundamental en la ***Criptografía Moderna*** son las **Funciones HASH**, que son usadas principalmente para resolver el problema de la Integridad de los mensajes, así como la Autenticidad de mensajes y de su origen.

Una **Función Hash** es también ampliamente usada para la **Firma Digital**, ya que los documentos a firmar pueden ser en general demasiado grandes. La **Función Hash** les asocia una cadena de longitud 160 bits que son mas manejables para el propósito de Firma Digital.
Esto es, un mensaje de longitud arbitraria lo transforma de forma única a un mensaje de longitud constante.

Veamos a continuación cómo hace esto teóricamente:

*La **Función Hash** toma como entrada una cadena de longitud arbitraria, digamos 5259 bits, luego divide este mensaje en pedazos iguales, digamos de 160 bits, como en este caso y en general el mensaje original no será un múltiplo de 160, entonces para completar un número entero de pedazos de 160 bits al último se le agrega un relleno, digamos de puros ceros. En nuestro caso en 5259 caben 32 pedazos de 160 bits y sobran 139, entonces se agregaran 21 ceros más.*
*Entonces el mensaje toma la forma **x=x1, x2, x3,...xt** donde cada **xi** tiene igual longitud (160 bits).*

Posteriormente se asocia un valor constante a un vector de inicialización IV, y se efectúan las siguientes interacciones:

$$H_0 = IV$$
$$H1 = f(H_{i-1}, X_i) \quad 1 <= i <= t$$
$$h(x) = g(H_t)$$

Figura 2.1

*Donde **f** es una función que combina a dos cadenas de bits de longitud igual y fija, y **g** es una función de salida.*

De alguna forma lo que se hace es tomar el mensaje, partirlo en pedazos de longitud constante y combinar de alguna forma pedazo por pedazo hasta obtener un solo mensaje de longitud fija como muestra la figura siguiente:

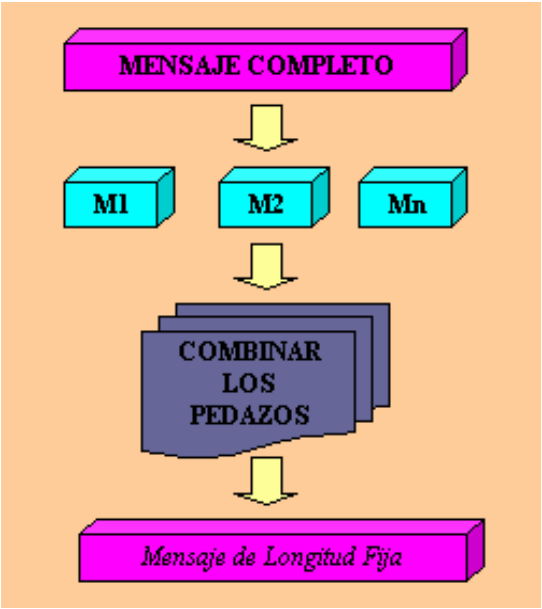


Figura 1.2

Las Funciones Hash (o primitivas hash) pueden operar como:

- **MDC (*Modification Deteccion Codes*)**

  Sirven para resolver el problema de la integridad de la información, al mensaje se le aplica un MDC (una función hash), y se manda junto con el propio mensaje, al recibirlo el receptor aplica la Función Hash al mensaje y comprueba que sea igual al hash que se envió antes.
  Los MDCs, son usados principalmente para resolver el problema de la Integridad y lo hacen tomando el razonamiento siguiente:

  - *Se aplica un hash* **h(M)** *al mensaje* **M** *y se envía con el mensaje, cuando se recibe* **(M, h(m))** *se le aplica una vez mas el hash (que es publica a* **M** *), obteniendo* **h'(m),** *si* **h(M)= h'(M),** *entonces se puede aceptar que el mensaje se a transmitido sin alteración.*

- **MAC (*Message Authentication Codes*)**

  Sirven para autenticar el orígen de los mensajes (junto con la integridad), un MAC es un mensaje junto con una clave simétrica que se les aplica un hash y se manda, al llegar la Autenticidad del orígen del mensaje se demuestra si la clave del receptor corresponde a la que se creó en el orígen del mensaje.
  Los MACs son usados para resolver el problema de autenticar el orígen del mensaje y se tiene el argumento siguiente:

  - *Se combina el mensaje* **M** *con una clave privada* **K** *y se les aplica un hash* **h(M,K)** *si al llegar a su destino* **h(M,K),** *se comprueba de Integridad de la clave privada* **K,** *entonces se demuestra que el orígen es solo el que tiene la misma clave* **K** *probando así la Autenticidad del orígen del mensaje.*

Entre las propiedades que rigen a las Funciones Hash tenemos:

X **Resistencia a la Preimagen:** Significa que dada cualquier imagen **y,** es computacionalmente imposible encontrar un mensaje **x,** tal que **h(x)=h(x)'.** Otra forma como se conoce esta propiedad es que **h** sea de *Un Solo Sentido.*

X **Resistencia a una Segunda Preimagen:** Significa que dado **x**, es computacionalmente imposible encontrar una **x'** tal que **h(x)=h(x)'.** Otra forma de conocer esta propiedad es que **h** sea resistente a una *Colisión Suave.*

X **Resistencia a Colisión:** Significa que computacionalmente es imposible encontrar dos diferentes mensajes **x, x'** tal que **h(x)=h(x)'.** Esta propiedad también se conoce como resistencia a *Colisión Fuerte.*

Las funciones **HASH** más conocidas y usadas son:

- **MD2:** Abreviatura de Message Digest 2, diseñado para ordenadores con procesador de 8 bits. Todavía se usa, pero no es recomendable, debido a su lentitud de proceso.

- **MD4:** Abreviatura de Message Digest 4, desarrollado por Ron Rivest, uno de los fundadores de RSA Data Security Inc. y padre del sistema asimétrico RSA. Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones hash. Un sistema de ataque desarrollado por Hans Dobbertin posibilita el crear mensajes aleatorios con los mismos valores de hash (colisiones), por lo que ya no se usa. De hecho, existe un algorítmo que encuentra una colisión en segundos.

- **MD5:** Abreviatura de Message Digest 5, también obra de Ron Rivest, que se creó para dar seguridad a MD4, y que ha sido ampliamente usado en diversos campos, como autenticador de mensajes en el protocolo SSL y como firmador de mensajes en el programa de correo PGP. Si embargo, fue reventado en 1996 por el mismo investigador que lo hizo con MD4, el señor Dobbertin, que consiguió crear colisiones en el sistema MD5, aunque por medio de ataques parciales. Pero lo peor es que también consiguió realizar ataques que comprometían la no-colisión, por lo que se podían obtener mensajes con igual hash que otro determinado. A pesar de todo esto, MD5 se sigue usando bastante en la actualidad.

- **SHA-1:** Secure Hash Algorithm, desarrollado como parte integral del Secure Hash Standard (SHS) y el Digital Signature Standard (DSS) por la Agencia de Seguridad Nacional Norteamericana, NSA.

Sus creadores afirman que la base de este sistema es similar a la de MD4 de Rivest, y ha sido mejorado debido a ataques nunca desvelados. La versión actual se considera segura (por lo menos hasta que se demuestre lo contrario) y es muy utilizada en algorítmos de firma, como en el programa PGP en sus nuevas claves DH/DSS (Diffie-Hellman/Digital Signature Standard). Destacar también que en la actualidad se están estudiando versiones de SHA con longitudes de clave de 256, 384 y 512 bits.

- **RIPEMD-160:** Desarrollado por un grupo de investigadores europeos, entre los que se encuentra Hans Dobbertin (el reventador de MD4-MD5) y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión tenía las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja claves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

2.2.3. INTEGRIDAD DE DATOS.

Ofrece el servicio de Integridad mediante el cifrado de una cadena comprimida de los datos.

2.2.4. FIRMA DIGITAL.

2.2.4.1. Introducción.

Desde el principio de las operaciones mercantiles, las personas han utilizado la firma manuscrita como medio para acreditar la identidad del firmante de un documento, dando su aprobación al contenido de lo firmado. Esta ha sido, y sigue siendo, ampliamente usada por las sociedades humanas desde hace siglos. Su equivalente en la actual sociedad de la informática es lo que se conoce como **Firma Digital,** que como su propio nombre indica, es la firma que acompaña a los documentos electrónicos y que demuestran la aprobación de su contenido por la persona que los envía.

Dando un significado más conceptual la **Firma Digital** o **Firma Electrónica** se puede definir como el conjunto de datos que se añaden a una unidad de datos para protegerlos contra la falsificación, permitiendo al receptor probar la fuente y la integridad de los mismos.

Una **Firma Digital** en un documento es el resultado de aplicar una **Función Hash** al documento. Para que sea de utilidad, la **Función Hash** necesita satisfacer dos propiedades importantes:

- **Primero**, debería ser difícil encontrar dos documentos cuyo valor para una función `hash' sea el mismo.

- **Segundo,** dado un valor hash debería ser difícil de recuperar el documento que produjo es valor.

Una **Firma Digital** certifica un documento y le añade una marca de tiempo. Si posteriormente el documento fuera modificado en cualquier modo, el intento de verificar la firma fallaría. La utilidad de una **Firma Digital** es la misma que la de una firma escrita a mano, sólo que la primera tiene una resistencia a la falsificación.

***Por ejemplo*:** La distribución del código fuente de algún software (GnuPG) viene firmado con el fin de que los usuarios puedan verificar que no ha habido ninguna manipulación o modificación al código fuente desde que fue archivado.

Algunos sistemas de cifrado de clave pública se pueden usar para firmar documentos. El firmante cifra el documento con su clave privada. Cualquiera que desee comprobar la firma y ver el documento, no tiene más que usar la clave pública del firmante para descifrarla. Este algorítmo satisface las dos propiedades necesarias para una buena función de **Hash**, pero en la práctica este algorítmo es demasiado lento para que sea de utilidad.

Al usar uno de estos algorítmos, un documento se firma con una **Función Hash**, y el valor del **Hash** es la firma. Otra persona puede comprobar la firma aplicando también una **Función Hash** a su copia del documento y comparando el valor **Hash** resultante con el del documento original. Si concuerdan, es casi seguro que los documentos son idénticos.

Claro que el problema está en usar una **Función Hash** para **Firmas Digitales** que no permita que un atacante interfiera en la comprobación de la firma. Si el documento y la firma se enviaran descifrados, un atacante podría modificar el documento y generar una firma correspondiente sin que lo supiera el destinatario. Si sólo se cifrara el documento, un atacante podría manipular la firma y hacer que la comprobación de ésta fallara.

Otra opción posible es usar un sistema de clave pública híbrido para cifrar tanto la firma como el documento. El firmante usa su clave pública, y cualquiera puede usar su clave pública para comprobar la firma y el documento.

La **Firma Digital** suele usarse en comunicaciones en las que no existe una confianza inicial total entre los comunicantes. Se usan para autentificar mensajes, para validar compras por Internet, para realizar transferencias de fondos bancarios y para otras transacciones de negocios.

Tanta es la fuerza que posee éste sistema que a nivel legal, la **Firma Electrónica** constituye en la mayoría de los casos una prueba de indudable de autoría del envío de un documento electrónico, semejante a la firma tradicional de puño y letra.

2.2.4.2. Proceso de Firma Digital.

El esquema básico de una **Firma Digital** básica sería el planteado a continuación:

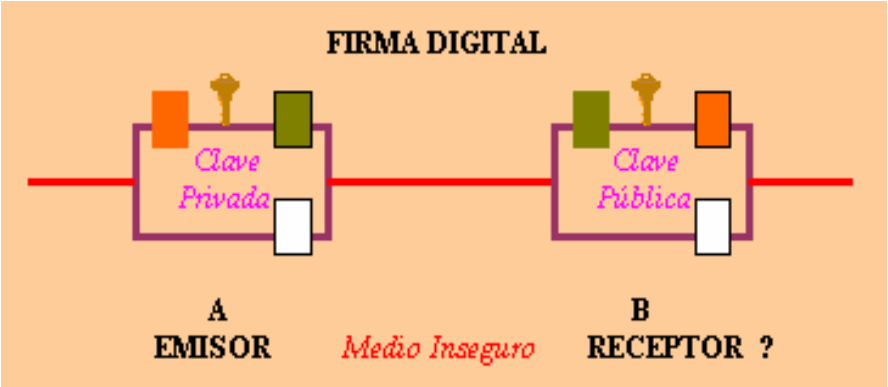


Figura 2.3

El proceso de **Firma Digital** consta de dos partes bien diferenciadas:

1. **Proceso de Firma**: El emisor encripta el documento con su llave privada, enviando al destinatario tanto el documento en claro como el encriptado.

2. **Proceso de Verificación de la Firma**: El receptor desencripta el documento cifrado con la clave pública del EMISOR (A) y comprueba que coincide con el documento original, lo que atestigua de forma total que el EMISOR (A), fue el que envío el mensaje.

El método de la **Firma Digital** no sólo proporciona Autenticidad al mensaje enviado por el EMISOR (A), si no que también asegura el No Repudio, ya que sólo el dueño de una llave privada puede encriptar un documento de tal forma que se pueda desencriptar con su llave pública, lo que garantiza que ha sido el EMISOR (A) y no otro el que ha enviado dicho documento.

2.2.4.3. Creación y Verificación de Firmas Digitales.

Para crear y verificar **Firmas Digitales**, se utiliza el par público y privado de claves en una operación que es diferente a la de cifrado y descifrado.
Se genera una firma con la clave privada del firmante. La firma se verifica por medio de la clave pública correspondiente.

Dicho de otra manera, el receptor descifra la firma con la clave pública del emisor, comprime con la función hash al texto original recibido y compara el resultado de la parte descifrada con la parte comprimida, si ambas coinciden el emisor tiene garantía de que el texto no ha sido modificado.

Como el emisor utiliza su clave secreta para cifrar la parte comprimida del mensaje, puede probarse ante una tercera parte, que la firma sólo ha podido ser generada por el usuario que guarda la clave secreta.

Por ejemplo: Antonio haría uso de su propia clave privada para firmar digitalmente la entrega de su última ponencia a la Revista de Informática.
El editor asociado que la recibiera, usaría la clave pública de Antonio para comprobar la firma, verificando de este modo que el envío proviene realmente de Antonio, y que no ha sido modificado desde el momento en que Antonio lo firmó.
Un alegato que podría evitar el EMISOR (A) para negar la autoría del envío de un documento cifrado con su clave privada sería el hecho de haber perdido dicha llave o que se la hayan sustraído, pero entonces hay que tener en cuenta que el EMISOR (A) es el única responsable del buen uso de su llave privada, por lo que está obligado a comunicar inmediatamente a la autoridad correspondiente cualquier circunstancia que ponga en peligro la seguridad de la misma.

Esto es parecido a lo que ocurre con las tarjetas de débito o crédito, siendo siempre en último extremo responsable del uso indebido de las mismas el dueño de la tarjeta si no ha avisado a tiempo a su entidad financiera o banco de la pérdida o sustracción.

2.2.4.4. Procedimiento de Firma.

Para describir como funciona el procedimiento de **Firma Digital,** lo haremos citando un ejemplo para su mejor comprensión:

a. Tony aplica una función resumen (HASH), **f** al mensaje **M**. El resultado **f(M)** es la huella digital del mensaje.
b. A continuación, Tony cifra con su clave privada este resultado.
c. Tony envía conjuntamente el mensaje y la firma digital (la huella digital cifrada con su clave privada) a Luis.
d. Luis, que conoce la función resumen utilizada y la clave pública de Tony, realiza dos operaciones: aplica la función resumen al mensaje y descifra la firma digital.
e. Si ambos resultados coinciden, Luis puede tener la certeza de dos hechos: que el mensaje no ha sido modificado en su tránsito por la red; y que el mensaje ha sido remitido, efectivamente, por Tony.

Lo anteriormente dicho es muy sencillo de comprender pero se presenta un problema que se tiene que ver con mucho cuidado:

¿Cómo aseguramos que el par clave pública - clave privada que asociamos a Tony es realmente suyo y no de un intruso?
La **Firma Electrónica** es individual a cada persona e intransferible, debe ser suministrada por un proveedor de servicios de certificación que cumpla con los

requisitos legales pertinentes y que garantice la Privacidad y viabilidad de dicha firma.
Los principios fundamentales de la firma electrónica están regidos por los Servicios de Seguridad (vistos anteriormente) como son: **Confidencialidad de los datos** (lo que da lugar a una pequeña y temporal red privada virtual sobre una red pública como Internet), **Autenticidad** (que certifica que la firma electrónica pertenece efectivamente a la persona que la envía), **Integridad** (que garantiza que los datos no han sido alterados desde el momento en que la persona les añadió la citada firma), **No repudio** (que supone que la firma electrónica fue añadida por su titular con intención de signar los datos, dando pleno consentimiento a su contenido).

2.2.4.5. Consecuencia del uso de Firmas Digitales.

Una consecuencia directa del uso de Firmas Digitales es la dificultad en negar que fue el propio usuario quien puso la firma digital, ya que ello implicaría que su clave privada ha sido puesta en peligro.

En resumen podemos decir que la **Firma Digital** en la vida real consiste en una cadena que contiene el resultado de cifrar básicamente con RSA aplicando la clave privada del firmante, una versión comprimida, mediante una función hash unidireccional y libre de colisiones, del texto a firmar.

2.2.5. CONTROL DE ACCESO.

Identificación y Autenticación de las entidades que intentan acceder, barreras de protección.

2.2.6. TRAFICO DE RELLENO.

Se envía tráfico de relleno para proteger frente a análisis de tráfico.

2.2.7. CONTROL DE ENCAMINAMIENTO.

Dirigir la información por zonas consideradas seguras y evitar zonas inseguras.

2.2.8. UNICIDAD.

Añadir a los datos una fecha, número de secuencia o número aleatorio para evitar ataques de alteración de orden o retención temporal de información.

2.3. ATAQUES DE SEGURIDAD.

2.3.1. INTRODUCCIÓN.

Antes de entrar a abordar el tema, cabe señalar algunos aspectos de suma importancia para la comprensión de lo que es un **Ataque Informático.**

Alguna vez nos hemos hecho esta pregunta, ***Qué queremos proteger?***

Tratando de responder la pregunta anterior, planteamos el siguiente análisis:
Los tres elementos principales a proteger en cualquier sistema informático son el ***software*, el *hardware* y los *datos***.

Por **hardware** entendemos el conjunto formado por todos los elementos físicos de un sistema informático (CPU's, terminales, cableado, medios de almacenamiento secundario cintas, CD-ROM, diskettes, o tarjetas de red).
Por **software** entendemos el conjunto de programas lógicos que hacen funcional al *hardware*, (sistemas operativos, aplicaciones).
Y por **datos** el conjunto de información lógica que manejan el *software* y el *hardware*, (paquetes que circulan por un cable de red o entradas de una base de datos). Aunque generalmente en las auditorías de seguridad se habla de un cuarto elemento a proteger, los **fungibles** que son elementos que se gastan o desgastan con el uso continúo, (papel de impresora, *tóners*, cintas magnéticas, *diskettes*).

Habitualmente los datos constituyen el principal elemento de los tres a proteger, ya que es el más amenazado y seguramente el más difícil de recuperar.
Ejemplo: Con toda seguridad una máquina UNIX está ubicada en un lugar de acceso físico restringido, o al menos controlado, y además en caso de pérdida de una aplicación (o un programa de sistema, o el propio núcleo de UNIX), este *software* se puede restaurar sin problemas desde su medio original (por ejemplo, el CD-ROM con el sistema operativo que se utilizó para su instalación). Sin embargo, en caso de pérdida de una base de datos o de un proyecto de un usuario, no tenemos un medio original desde el que restaurar; hemos de pasar obligatoriamente por un sistema de copias de seguridad, y a menos que la política de copias sea muy estricta, es difícil devolver los datos al estado en que se encontraban antes de la pérdida.

Contra cualquiera de los tres elementos descritos anteriormente (pero principalmente sobre los datos), se pueden realizar multitud de ataques o, dicho de otra forma, están expuestos a diferentes amenazas.

Seguidamente, y dándole continuidad a la pregunta anterior, otra posible pregunta que nos plantearíamos sería, ***De qué o quién nos queremos proteger?***
En la gran mayoría de publicaciones relativas a la seguridad informática en general, tarde o temprano se intenta clasificar en grupos a los posibles elementos que pueden atacar nuestro sistema.

Con frecuencia, especialmente en las obras menos técnicas y más orientadas a otros aspectos de la seguridad, se suele identificar a los atacantes únicamente

como personas; esto tiene sentido si hablamos por ejemplo de responsabilidades por un delito informático.

Pero en este caso es preferible hablar de elementos y no de personas en general; aunque a veces lo olvidemos, nuestro sistema puede verse perjudicado por múltiples entidades aparte de humanos, como por ejemplo programas, catástrofes naturales o, por qué no, fuerzas extraterrestres; si un usuario pierde un trabajo importante a causa de un ataque, poco le importará que haya sido un intruso, un gusano, un simple error del administrador, o un *alien* que haya afectado un disco duro.

A continuación, presentamos una relación de los elementos que potencialmente pueden amenazar a nuestro sistema. Por su parte con esto no se pretende dar una clasificación exhaustiva, ni por supuesto una taxonomía formal, simplemente tratamos de proporcionar una idea acerca de qué o quién amenaza a un Sistema Informático.

- **PERSONAS.**

  La mayoría de ataques a nuestro sistema van a provenir en última instancia de personas que, intencionada o inintencionadamente, pueden causarnos enormes pérdidas. Generalmente estas personas se tratarán de piratas que intentan conseguir el máximo nivel de privilegio posible aprovechando alguno (o algunos) de los riesgos lógicos, especialmente agujeros del *software*. Pero con demasiada frecuencia se suele olvidar que los "piratas clásicos" no son los únicos que amenazan nuestros equipos; es especialmente preocupante que mientras que hoy en día cualquier administrador mínimamente preocupado por la seguridad va a conseguir un sistema relativamente fiable de una forma lógica (permaneciendo atento a vulnerabilidades de su *software*, restringiendo servicios, utilizando cifrado de datos...), pocos administradores tienen en cuenta factores como la ingeniería social o el basureo a la hora de diseñar una Política de Seguridad.

  Pasemos entonces a describir brevemente los diferentes tipos de personas que de una u otra forma pueden constituir un riesgo para nuestros sistemas. Generalmente los curiosos y los *crackers* realizan ataques pasivos (que se pueden convertir en activos), mientras que los terroristas y ex-empleados realizan ataques activos puros; los intrusos remunerados suelen ser atacantes pasivos si nuestra red o equipo no es su objetivo, y activos en caso contrario, y el personal realiza ambos tipos indistintamente, dependiendo de la situación concreta.

  Personal:

  Las amenazas a la seguridad de un sistema que provienen del personal de la propia organización rara vez son tomadas en cuenta; se presume un entorno de confianza donde a veces no existe, por lo que se pasa por alto el hecho de que casi cualquier persona de la

organización, incluso el personal ajeno a la infraestructura informática (secretariado, personal de seguridad, personal de limpieza y mantenimiento...) puede comprometer la seguridad de los equipos.
Aunque los ataques pueden ser intencionados (en cuyo caso sus efectos son extremadamente dañinos, recordemos que nadie mejor que el propio personal de la organización conoce mejor los sistemas y sus debilidades), lo normal es que más que de ataques se trate de **accidentes** causados por un error o por desconocimiento, de las normas básicas de seguridad.

Un empleado de mantenimiento que corta el suministro eléctrico para hacer una reparación puede llegar a ser tan peligroso como el más experto de los administradores que se equivoca al teclear una orden y borra todos los sistemas de ficheros; y en el primer caso, el atacante ni siquiera ha de tener acceso lógico (ni físico) a los equipos, ni conocer nada sobre seguridad en un Sistema. Hemos de recordar siempre que decir *"No lo hice a propósito"* no va a servir para recuperar datos perdidos ni para restaurar un *hardware* dañado o robado.

Ex-empleados:

Otro gran grupo de personas potencialmente interesadas en atacar nuestro sistema son los antiguos empleados del mismo, especialmente los que no abandonaron el entorno por voluntad propia (y en el caso de redes de empresas, los que pasaron a la competencia). Generalmente, se trata de personas descontentas con la organización que pueden aprovechar debilidades de un sistema que conocen perfectamente para dañarlo como venganza por algún hecho que no consideran justo. Amparados en excusas como *`No me han pagado lo que me deben'* o *`Es una gran universidad, se lo pueden permitir'* pueden insertar troyanos, bombas lógicas, virus o simplemente conectarse al sistema como si aún trabajaran para la organización (muchas veces se mantienen las cuentas abiertas incluso meses después de abandonar la universidad o empresa), conseguir el privilegio necesario, y dañarlo de la forma que deseen, incluso chantajeando a sus ex-compañeros o ex-jefes.

Curiosos:

Junto con los *crackers*, los curiosos son los atacantes más habituales de un Sistema en Redes. Recordemos que los equipos están trabajando en entornos donde se forma a futuros profesionales de la informática y las telecomunicaciones (gente que *a priori* tiene interés por las nuevas tecnologías), además recordemos también que las personas suelen ser curiosas por naturaleza; ésta combinación produce una avalancha de estudiantes o personal intentando conseguir mayor privilegio del que tienen o intentando acceder a sistemas a los que oficialmente no tienen acceso. Y en la mayoría de

ocasiones esto se hace simplemente para leer el correo de un amigo, enterarse de cuánto cobra un compañero, copiar un trabajo o comprobar que es posible romper la seguridad de un sistema concreto.

Aunque en la mayoría de situaciones se trata de ataques no destructivos (a excepción del borrado de huellas para evitar la detección), parece claro que no benefician en absoluto al entorno de fiabilidad que podamos generar en un determinado sistema.

Crackers:

Los entornos de seguridad media son un objetivo típico de los intrusos, ya sea para espiar, para utilizarlas como enlace hacia otras redes o simplemente por diversión. Por un lado, son redes generalmente abiertas, y la seguridad no es un factor tenido muy en cuenta en ellas; por otro lado, el gran número y variedad de Sistemas conectados a estas redes provoca, casi por simple probabilidad, que al menos algunos de sus equipos (cuando no la mayoría) sean vulnerables a problemas conocidos de antemano. De esta forma un atacante sólo ha de utilizar un escáner de seguridad contra el dominio completo y luego atacar mediante un simple *exploit* los equipos que presentan vulnerabilidades; esto convierte a las redes de I+D, a las de empresas, o a las de ISPs en un objetivo fácil y apetecible para piratas con cualquier nivel de conocimientos, desde los más novatos (y a veces más peligrosos) hasta los expertos, que pueden utilizar toda la red para probar nuevos ataques o como nodo intermedio en un ataque a otros organismos, con el consiguiente deterioro de imagen (y a veces de presupuesto) que supone para una universidad ser, sin desearlo, un apoyo a los piratas que atacan sistemas teóricamente más protegidos, como los militares.

Terroristas:

Por terroristas no debemos entender simplemente a los que se dedican a poner bombas o quemar autobuses u otras actividades similares, sino también que ésta definición engloba a cualquier persona que ataca al sistema simplemente por causar algún tipo de daño en él.

Por ejemplo: Alguien puede intentar borrar las bases de datos de un partido político enemigo o destruir los sistemas de ficheros de un servidor que alberga páginas *web* de algún grupo religioso; en el caso de redes de I+D, típicos ataques son la destrucción de sistemas de prácticas o la modificación de páginas *web* de algún departamento o de ciertos profesores, generalmente por parte de alumnos descontentos.

Intrusos Remunerados

Este es el grupo de atacantes de un sistema más peligroso, aunque por fortuna el menos habitual en redes normales; suele afectar más a las grandes (muy grandes), empresas o a organismos de defensa. Se trata de piratas con gran experiencia en problemas de seguridad y un amplio conocimiento del sistema, que son pagados por una tercera parte generalmente para robar secretos (el nuevo diseño de un procesador, una base de datos de clientes, información confidencial sobre las posiciones de satélites espía) o simplemente para dañar la imagen de la entidad afectada. Esta tercera parte suele ser una empresa de la competencia o un organismo de inteligencia, es decir, una organización que puede permitirse un gran gasto en el ataque; de ahí su peligrosidad.
Se suele pagar bien a los mejores piratas, y por si esto fuera poco los atacantes van a tener todos los medios necesarios a su alcance.

Aunque como hemos dicho los intrusos remunerados son los menos comunes en la mayoría de situaciones, en ciertas circunstancias pueden aprovechar nuestras redes como plataforma para atacar otros organismos. Para poner un ejemplo de la situación anterior veamos el siguiente caso de la vida real, en la que el experto en seguridad Cliff Stoll describe cómo piratas pagados por la KGB soviética utilizaron redes y sistemas UNIX dedicados a I+D para acceder a organismos de defensa e inteligencia estadounidenses.

❖ AMENAZAS LÓGICAS.

Bajo el seudónimo de Amenazas Lógicas encontramos todo tipo de programas que de alguna forma pueden dañar a un Sistema, creados de forma intencionada para ello (*software* malicioso, también conocido como *malware*) o simplemente por error (*bugs* o agujeros).

Software Incorrecto:

Las amenazas más habituales a un Sistema provienen de errores cometidos de forma involuntaria por los programadores de sistemas o de aplicaciones.
A estos errores de programación se les denomina *bugs*, y a los programas utilizados para aprovechar uno de estos fallos y atacar al sistema, *exploits*.
Como hemos dicho, representan la amenaza más común contra un Sistema, especialmente a UNIX, ya que cualquiera puede conseguir un *exploit* y utilizarlo contra nuestra máquina sin ni siquiera saber cómo funciona y sin unos conocimientos mínimos de UNIX. Incluso hay *exploits* que dañan seriamente la integridad de un sistema y están preparados para ser utilizados desde MS-DOS, con lo que cualquier pirata novato (comúnmente, se les denomina *Script Kiddies*) puede utilizarlos contra un servidor y conseguir un control total de una máquina que cueste mucho

dinero desde su PC sin saber nada del sistema atacado; incluso hay situaciones en las que se analizan los *logs* de estos ataques y se descubre que el pirata incluso intenta ejecutar órdenes de MS-DOS.

❖ HERRAMIENTAS DE SEGURIDAD.

Cualquier herramienta de seguridad representa un arma de doble filo, de la misma forma que un administrador las utiliza para detectar y solucionar fallos en sus sistemas o en la subred completa. Un potencial intruso las puede utilizar para detectar esos mismos fallos y aprovecharlos para atacar los equipos. Herramientas como NESSUS, SAINT o SATAN pasan de ser útiles a ser peligrosas cuando las utilizan *crackers* que buscan información sobre las vulnerabilidades de un *host* o de una red completa.
La distribución libre de herramientas que puedan facilitar un ataque es un tema peliagudo y muy criticado por el cual han recibido enormes críticas por diseñar determinadas herramientas de seguridad. Tras numerosos debates sobre el tema, ha quedado bastante claro que no se puede basar la seguridad de un sistema en el supuesto desconocimiento de sus problemas por parte de los atacantes.
Esta política, denominada *Security Through Obscurity*, se ha demostrado inservible en múltiples ocasiones. Si como administradores no utilizamos herramientas de seguridad que muestren las debilidades de nuestros sistemas (para corregirlas), tenemos que estar seguro que un atacante no va a dudar en utilizar tales herramientas (para explotar las debilidades encontradas); por tanto, hemos de agradecer a los diseñadores de tales programas el esfuerzo que han realizado (y nos han ahorrado) en pro de sistemas más seguros.

❖ PUERTAS TRASERAS.

Durante el desarrollo de aplicaciones grandes o de sistemas operativos es habitual entre los programadores insertar atajos o desvíos en los sistemas habituales de autenticación del programa o del núcleo que se está diseñando. A estos atajos se les denomina puertas traseras (backdoors), y con ellos se consigue mayor velocidad a la hora de detectar y depurar fallos:
Por ejemplo: Los diseñadores de un *software* de gestión de bases de datos en el que para acceder a una tabla se necesiten cuatro claves diferentes de diez caracteres cada una pueden insertar una rutina para conseguir ese acceso mediante una única clave especial, con el objetivo de perder menos tiempo al depurar el sistema.

Algunos programadores pueden dejar estos atajos en las versiones definitivas de su *software* para facilitar un mantenimiento posterior, para garantizar su propio acceso, o simplemente por descuido; la cuestión es que si un atacante descubre una de estas puertas traseras (no nos importa el método que utilice para hacerlo) va a tener un acceso global a datos que no debería poder leer, lo que obviamente supone un grave peligro para la integridad de nuestro sistema.

❖ BOMBAS LÓGICAS.

Las bombas lógicas son partes de código de ciertos programas que permanecen sin realizar ninguna función hasta que son activadas; en ese punto, la función que realizan no es la original del programa, sino que generalmente se trata de una acción perjudicial.
Los activadores más comunes de estas bombas lógicas pueden ser la ausencia o presencia de ciertos ficheros, la ejecución bajo un determinado UID (contraseñas) o la llegada de una fecha concreta; cuando la bomba se activa va a poder realizar cualquier tarea que pueda realizar la persona que ejecuta el programa. Si las activa el *root*, o el programa que contiene la bomba está setuidado a su nombre, los efectos obviamente pueden ser fatales.

❖ CANALES CUBIERTOS.

Según la definición, los canales cubiertos (o canales ocultos, según otras traducciones) son canales de comunicación que permiten a un proceso transferir información de forma que viole la Política de Seguridad del sistema; dicho de otra forma, un proceso transmite información a otros (locales o remotos) que no están autorizados a leer dicha información.

Los canales cubiertos no son una amenaza demasiado habitual en redes de I+D, ya que suele ser mucho más fácil para un atacante aprovechar cualquier otro mecanismo de ataque lógico; sin embargo, es posible su existencia, y en este caso su detección suele ser difícil. Algo tan simple como el puerto finger abierto en una máquina puede ser utilizado a modo de *covert channel* por un pirata con algo de experiencia.

❖ VIRUS.

Un virus es una secuencia de código que se inserta en un fichero ejecutable (denominado *huésped*), de forma que cuando el archivo se ejecuta, el virus también lo hace, insertándose a sí mismo en otros programas.
Todo el mundo conoce los efectos de los virus en algunos sistemas operativos de sobremesa; sin embargo, en el Sistema Operativo UNIX, los virus no suelen ser un problema de seguridad grave, ya que lo que pueda hacer un virus lo puede hacer más fácilmente cualquier otro mecanismo lógico (que será el que hay que tener en cuenta a la hora de diseñar una Política de Seguridad).
Aunque los virus existentes para entornos UNIX son más una curiosidad que una amenaza real, en sistemas sobre plataformas IBM-PC o compatibles (recordemos que hay muchos sistemas UNIX que operan en estas plataformas, como Linux, FreeBSD, NetBSD, Minix, Solaris) ciertos virus, especialmente los de *boot*, pueden tener efectos nocivos, como dañar el sector de arranque; aunque se trata de daños menores comparados con los efectos de otras amenazas, hay que tenerlos en cuenta.

Gusanos:

Un gusano es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando *bugs* de los sistemas a los que conecta para dañarlos. Al ser difíciles de programar su número no es muy elevado, pero el daño que pueden causar es muy grande. El mayor incidente de seguridad en Internet fue precisamente el *Internet Worm*, un gusano que en 1988 causó perdidas millonarias al infectar y detener más de 6000 máquinas conectadas a la red.

Hemos de pensar que un gusano puede automatizar y ejecutar en unos segundos todos los pasos que seguiría un atacante humano para acceder a nuestro sistema. Mientras que una persona, por muchos conocimientos y medios que posea, tardaría como mínimo horas en controlar nuestra red completa (un tiempo más que razonable para detectarlo), un gusano puede hacer eso mismo en pocos minutos; de ahí su enorme peligro y sus devastadores efectos.

Caballos de Troya:

Los troyanos o caballos de Troya son instrucciones escondidas en un programa de forma que éste parezca realizar las tareas que un usuario espera de él, pero que realmente ejecute funciones ocultas (generalmente en detrimento de la seguridad) sin el conocimiento del usuario; como el Caballo de Troya de la mitología griega, al que deben su nombre, ocultan su función real bajo la apariencia de un programa inofensivo que a primera vista funciona correctamente.

Por ejemplo: Es típico utilizar lo que se denomina un *rootkit*, que no es más que un conjunto de versiones troyanas de ciertas utilidades (netstat, ps, who), para conseguir que cuando el administrador las ejecute no vea la información relativa al atacante, como sus procesos o su conexión al sistema.

Otro programa que se suele suplantar es login:

Por ejemplo: Para que al recibir un cierto nombre de usuario y contraseña proporcione acceso al sistema sin necesidad de consultar /etc/passwd.

Programas conejo o bacterias:

Bajo este nombre se conoce a los programas que no hacen nada útil, sino que simplemente se dedican a reproducirse hasta que el número de copias acaba con los recursos del sistema (memoria, procesador, disco...), produciendo una negación de servicio. Por sí mismos no hacen ningún daño, sino que lo que realmente perjudica es el gran número de copias suyas en el sistema, que en algunas situaciones pueden llegar a provocar la parada total de la máquina.

Hemos de pensar hay ciertos programas que pueden actuar como conejos sin proponérselo;

Ejemplo: Se suelen encontrar en los sistemas UNIX destinados a prácticas en las que se enseña a programar al alumnado: es muy

común que un bucle que por error se convierte en infinito contenga entre sus instrucciones algunas de reserva de memoria, lo que implica que si el sistema no presenta una correcta política de cuotas para procesos de usuario pueda venirse abajo o degradar enormemente sus prestaciones. El hecho de que el autor suela ser fácilmente localizable no debe ser ninguna excusa para descuidar esta política. No podemos culpar a un usuario por un simple error, y además el daño ya se ha producido.

❖ TÉCNICAS SALAMI.

Por Técnica Salami se conoce al robo automatizado de pequeñas cantidades de bienes (generalmente dinero) de una gran cantidad orígen. El hecho de que la cantidad inicial sea grande y la robada pequeña hace extremadamente difícil su detección.

Por ejemplo: Si de una cuenta con varios millones de dólares se roban unos céntimos, nadie va a darse cuenta de ello; si esto se automatiza para descontar un dólar de cada nómina pagada en la universidad o de cada beca concedida, tras un mes de actividad seguramente se habrá robado una enorme cantidad de dinero sin que nadie se haya percatado de este hecho, ya que de cada orígen se ha tomado una cantidad ínfima.

Las técnicas salami no se suelen utilizar para atacar sistemas normales, sino que su uso más habitual es en sistemas bancarios; sin embargo, como en una red con requerimientos de seguridad medios es posible que haya ordenadores dedicados a contabilidad, facturación de un departamento o gestión de nóminas del personal, comentamos esta potencial amenaza contra el *software* encargado de estas tareas.

❖ CATÁSTROFES.

Las catástrofes (naturales o artificiales) son la amenaza menos probable contra los entornos habituales, simplemente por su ubicación geográfica, a nadie se le escapa que la probabilidad de sufrir un terremoto o una inundación que afecte a los sistemas informáticos en una gran ciudad. Este hecho es relativamente bajo, al menos en comparación con el riesgo de sufrir un intento de acceso por parte de un pirata o una infección por virus. Sin embargo, el hecho de que las catástrofes sean amenazas poco probables no implica que contra ellas no se tomen algunas medidas básicas, ya que si se produjeran generarían los mayores daños.

Un subgrupo de las catástrofes es el denominado de **Riesgos Poco Probables**. Obviamente se denomina así al conjunto de riesgos que, aunque existen, la posibilidad de que se produzcan es tan baja (menor incluso que la del resto de catástrofes) que nadie toma, o nadie puede tomar, medidas contra ellos.

Por ejemplo: Un ejemplo de Riesgos Poco Probables es un ataque nuclear contra el sistema, el impacto de un satélite contra la sala de operaciones, o la

abducción de un operador por una nave extraterrestre. Nada nos asegura que este tipo de catástrofes no vaya a ocurrir, pero la probabilidad es tan baja y los sistemas de prevención tan costosos que no vale la pena tomar medidas contra ellas.

Por ejemplo: Un ejemplo de catástrofes hablaremos de terremotos, inundaciones, incendios, humo o atentados de baja magnitud (más comunes de lo que podamos pensar); obviamente los riesgos poco probables los trataremos como algo anecdótico o superficial. De cualquier forma, vamos a hablar de estas amenazas sin extendernos mucho, ya que el objetivo de este tema es solamente dar a conocer los eventos de los que se debe de proteger un Sistema.

Una vez abordados y explicados algunos de factores o problemas que enfrenta un Sistema Informático pasaremos a abordar la problemática en si de los **Ataques de Seguridad.**

Debemos de entender como **Ataque de Seguridad** a cualquier acción que comprometa la seguridad de la información propietaria de una organización o empresa. Por ejemplo un intento de Criptoanálisis es llamado un ataque.

A continuación se describen dos categorías generales de ataques:

2.3.2. ATAQUES PASIVOS.

Estos ataques son en los que la persona que se entromete no altera la comunicación, sino que la escucha o monitoriza, así pues puede saber el orígen de la comunicación y destinatario mirando las cabeceras de los paquetes, saber cual es el volumen de tráfico y que horas son las habituales de comunicación entres entidades. Todos estos ataques pasivos son difíciles de interceptar ya que no provocan alteración de los datos, pero sí lo podemos evitar mediante el cifrado de la información.

La amenaza activa contra un mensaje secreto es su robo o el acceso no autorizado. El éxito del oponente es obtener información que se transmite.
Para defenderse de este tipo de ataques hay que abocarse a la prevención más que a la detección. Dentro de este tipo de ataque podemos clasificar los siguientes:

- ***Interrupción.***

Este tipo de ataque, inhabilita una parte del sistema impidiendo su funcionamiento (formatea disco duro, reinicia el sistema operativo, hace caer algún enlace...). Esto quiere decir que un valor del sistema se destruye o no está disponible. Esto es entonces un ataque sobre la Disponibilidad.

- ***Interceptación.***

Este ataque, hace que una parte no autorizada acceda a un valor o información confidencial (puede esta en la red). Este es un ataque sobre la Confidencialidad. La parte no autorizada podría ser una persona, un programa, o un computador.
También puede ser interesante un simple análisis del trafico, sin importar el contenido.

2.3.3 ATAQUES ACTIVOS.

Estos ataques son los que se realizan algún tipo de modificación de los datos transmitidos, o bien crean un flujo falso de datos, suplantando la identidad de la entidad emisora, reactuando, es decir de otra manera, uno o varios de los mensajes buenos son reenviados varias veces, modificando el mensaje no en su totalidad sino en parte, produciéndose la degradación del servicio para no permitir el uso normal de la comunicación o paralizando el servicio temporalmente.
La amenaza activa, contra un mensaje secreto es su alteración, estos presentan características opuestas a las de los pasivos.

Es difícil prevenir absolutamente a los ataques activos porque se requiere protección física de todas las facilidades y trayectorias de las comunicaciones en todo momento. En lugar de eso, el objetivo es detectar los ataques activos y recuperarlos de cualquier retraso que causen.
En cuanto a este tipo de ataques caben citar:

- ***Modificación.***

En este ataque, una parte no autorizada no solamente accede sino que manipula indebidamente un valor. Este es un ataque sobre la Integridad. Dicho de otra forma cierta información es modificada por una parte no autorizada.

- ***Fabricación.***

Este ataque actúa sobre la Autenticidad. Una parte no autorizada inserta objetos falsificados dentro del sistema.

Una comunicación, protegida o no, mediante sistemas criptográficos, está sujeta a una gran variedad de ataques de los cuales es imposible dar una taxonomía completa.

Los citados a continuación son los que se presentan habitualmente y con más frecuencia desde el punto de vista de los Criptoanalistas:

⇨ ***Ataque solo al Criptograma***: Es el más desfavorable para el intruso o Criptoanalista. En este caso, sólo tiene acceso al texto cifrado. El trabajo del intruso consiste en recuperar el texto en claro de tantos mensajes como sea posible. En tales condiciones, y aunque conociera el algorítmo de cifrado, sólo puede intentar vulnerar dicho algorítmo, realizar un análisis estadístico de los criptogramas o probar todas las claves posibles del algorítmo. Este último caso, por motivos obvios se conoce como **"Búsqueda Exhaustiva"** o también como **"Ataque basado en Fuerza Bruta"**.

⇨ ***Ataque mediante Texto en Claro Conocido:*** Este ataque, es más ventajoso para el atacante. Se ha hecho con pares de texto en claro y su equivalente cifrado o ha adivinado, de algún modo, el contenido del mensaje (muchos mensajes cifrados, correspondientes a protocolos normalizados, reproducen la misma estructura o poseen las mismas palabras en los mismos sitios del mensaje). Estas parejas pueden ser usadas para llevar a cabo el Criptoanálisis y averiguar la clave, lo cual será útil si se usa la misma clave para posteriores comunicaciones.

⇨ ***Ataque mediante Texto en Claro Escogido:*** Es un ataque mucho más eficaz que el anterior, en el que el intruso es capaz, de algún modo, de conseguir que un texto elegido por él sea cifrado con la clave desconocida. Por tanto hay que diseñar el sistema criptográfico de modo que nunca un intruso pueda introducir mensajes propios.

⇨ ***Ataque Adaptable mediante Texto en Claro Escogido:*** Es un caso especial del anterior en el que el intruso no sólo puede elegir el texto que quiere cifrar, sino que puede tomar decisiones sobre el texto que ser cifra basadas en resultados anteriores.

⇨ ***Ataque mediante Criptogramas Escogidos:*** el atacante puede obtener el descifrado de diversos mensajes cifrados escogidos por él.

⇨ ***Criptoanálisis «rubber-hose»:*** El criptoanalista chantajea o tortura hasta que le den la clave. El soborno se refiere a un ataque de compra de la clave.
Estos son ataques muy poderosos y frecuentemente son el mejor modo de romper el algorítmo.

Se pueden diseñar otros tipos de ataques Criptoanalíticos, los citados a continuación son los más frecuentes en el marco de una comunicación entre dos entidades, además pueden servir para implementar alguno de los anteriores:

⇨ ***Escucha Pasiva* (*Passive Eavesdropping*):** El intruso simplemente escucha el tráfico que circula por el canal.

⇨ ***Tercero Interpuesto (man-in-the-middle):*** El intruso, de alguna forma, se coloca entre los dos interlocutores y hace creer a cada uno de ellos que es su interlocutor.

⇨ ***Retransmisión Ciega* (*Replay*):** El intruso intercepta un mensaje legítimo, lo almacena (sin eliminarlo) y lo reenvía un tiempo después.

⇨ ***Cortado y Pegado (cut-and-paste):*** Dados dos mensajes cifrados con la misma clave, a veces es posible combinar partes de los dos para producir uno nuevo. El intruso no sabe lo que dice este nuevo mensaje, pero puede utilizarlo para confundir a los interlocutores legítimos e inducir a alguno de ellos a hacer algo beneficioso para el intruso.

⇨ ***Puesta a Cero del reloj (time-resetting):*** En protocolos que utilizan de alguna forma la hora actual, el intruso puede tratar de confundirnos acerca de cuál es la verdadera hora.

⇨ ***Ataque con Clave Escogida.*** Este ataque no significa que el criptoanalista puede elegir la clave; significa que tiene algún conocimiento acerca de la relación entre diferentes claves. Es un ataque poco práctico.

Capítulo

3

CRIPTOSISTEMAS MODERNOS

3.1. CRIPTOGRAFÍA SIMÉTRICA O DE CLAVE SECRETA.

3.1.1. INTRODUCCIÓN.

La ***Criptografia Simétrica*** ha sido la más usada en toda la historia, ésta a podido ser implementada en diferentes dispositivos, manuales, mecánicos, eléctricos, hasta los algorítmos actuales que son programables en cualquier computadora.

La ***Criptografía Simétrica*** o de ***Clave Pública*** se refiere al conjunto de métodos que permiten tener comunicación segura entre dos partes Emisor y Receptor, por un canal inseguro siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos Clave Simétrica o Unica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar o sea se utiliza una clave secreta solo conocida por los dos interlocutores.

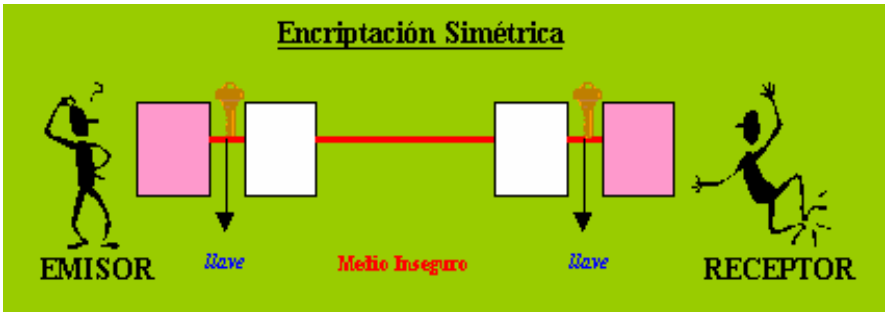


Figura 3.1

Este tipo de ***Criptografía*** es conocida también como Criptografia de Clave Privada o de Llave Privada.

Los cifradores simétricos pueden dividirse en dos grupos:

- **Cifradores de Flujo**, los cuales cifran un único bit del texto en claro cada vez.
- **Cifradores de Bloque**, que toman un grupo de bits (un valor típico es 64 bits) y lo cifran como si se tratase de una unidad.

La tendencia de los sistemas de Clave Simétrica, actualmente, es a utilizarlos poco o simplemente para cuestiones que no necesiten un alto grado de protección.
Estos tipos de CriptoSistemas están basados en los métodos clásicos de operaciones con bloques (datos y clave):

- Permutaciones (cajas P): cambio de orden en los bits.
- Sustituciones (cajas S): codificación de agrupaciones de bits.
- Operaciones *OR* exclusivo de grupos de bits con otros obtenidos de una clave.

Estos procesos se repiten en varias etapas en secuencia para aumentar la complejidad de la labor del criptoanalista.

Algunos algorítmos: **DES, Blowfish, CAST – 128, Crab, Feal, Khafre, Safer, Safer – SK, Triple DES, IDEA.**

3.1.2. CLASIFICACIÓN.

Entre los tipos de ***Criptografia Simétrica*** podemos citar la siguiente clasificación en tres familias:

- ***Criptografía Simétrica de Bloques (Block cipher):*** Aunque con ligeras modificaciones un sistema de Criptografía Simétrica de Bloques puede modificarse para convertirse en alguna de las otras dos formas, e inversamente, sin embargo es importante verlas por separado dado que se usan en diferentes aplicaciones.

- ***Criptografía Simétrica de Lluvia (Stream cipher):*** Los cifradores de lluvia o Stream Cipher, son usados donde se cuenta con un ancho de banda restringido (él numero de bits que se transmiten a la vez), además de que se requiere independencia en los bloques transmitidos, entonces la mejor opción es cifrar bit por bit o byte por byte. Este tipo de cifradores tiene la característica además de ser muy rápido.
  Los algorítmos más conocidos de este tipo están: RC-4, SEAL y WAKE.

- ***Criptografía Simétrica de Resumen (Hash Function):*** Descrita anteriormente *(Capitulo 2: Herramientas de Seguridad, pag 27 – 31).*

3.1.3. ALGORÍTMO DES (*DATA ENCRYPTION STANDARD*).

Es un algorítmo desarrollado en 1977 por el Departamento de Comercio y la Oficina Nacional de Estándares de EEUU en colaboración con la empresa IBM, a requerimiento del NBS (*National Bureau of Standards*, en la actualidad denominado NIST, *National Institute of Standards and Technology*) de EE.UU; y posteriormente modificado, adoptado y sometido a las leyes de Estados Unidos, pasando a formar parte de la ISO.

Fue creado y asignado como estándar de cifrado, con objeto de proporcionar al público en general un algorítmo de cifrado normalizado para redes de ordenadores y todas las informaciones sensibles no clasificadas.

Es el más estudiado y utilizado de los algorítmos de clave simétrica (o de cualquier otro tipo). Posteriormente, en 1980, el NIST estandarizó los diferentes modos de operación del algorítmo.

El nombre original del algorítmo, tal como lo denominó IBM, era ***Lucifer***. Trabajaba sobre bloques de 128 bits, teniendo la clave igual longitud. Se basaba en operaciones lógicas booleanas y podía ser implementado fácilmente, tanto en *software* como en *hardware*.

Tras las modificaciones introducidas por el NBS, consistentes básicamente en la reducción de la longitud de claves y bloques, **DES** cifra bloques de 64 bits cada vez, produciendo así 64 bits cifrados. El algorítmo, que se parametriza mediante una clave de 64 bits, de los que 8 son de paridad (esto es, 56 bits), tiene 19 etapas diferentes.

3.1.3.1. Descripción de DES.

- **ENTRADA:** Bloque de texto en claro de 64 bits, clave **K** de 64 bits (56 + 8 detección de errores).
- Una permutación inicial de los 64 bits de entrada.
- 16 iteraciones en intermedias (**Permutación**, **Sustitución** y **OR**), entre datos y subclaves obtenidas a partir de **K**.
- Una permutación inversa con los 64 bits de salida.
- **SALIDA:** bloque de texto cifrado de 64 bits.

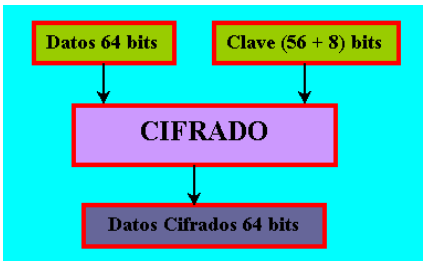


Figura 3.2

Se basa en un sistema monoalfabético, con un algorítmo de cifrado consistente en la aplicación sucesiva de varias permutaciones (transposición) y sustituciones los cuales hacen que sea particularmente difícil de romper.

Sin embargo, **DES** depende, tanto del que envía el mensaje como el que lo recibe y que estos conozcan la clave con la cual fue encriptada (la misma clave para los dos), y en este sentido se parece al sistema usado por los espartanos, que necesitaban tener el cilindro con el cual se había codificado el texto para poder leerlo.

En el caso de **DES** al "cilindro" se le denomina "llave". La seguridad de esta llave va a depender de su tamaño. Cuando tenemos un mensaje cifrado hay un número "n" de posibilidades de descubrir la llave con la cual sé encriptó. Así, la confiabilidad de una llave depende de que ese número "n" sea tan alto que a un atacante le tome demasiado tiempo probar todas las posibilidades. Una llave de 56 bits es actualmente el standard en **DES**.

Para leer un mensaje cifrado con **DES** es necesario usar la misma llave con la cual se encriptó, lo cual lo hace poco práctico e inseguro en el caso de transacciones comerciales virtuales, porque la propia llave debería transmitirse por medios electrónicos.

Inicialmente el texto en claro a cifrar se somete a una permutación, con bloque de entrada de 64 bits (o múltiplo de 64), para posteriormente ser sometido a la acción de dos funciones principales:

a. Una función de permutación con entrada de 8 bits.
b. Otra de sustitución con entrada de 5 bits, en un proceso que consta de 16 etapas de cifrado.

En general, **DES** utiliza una clave simétrica de 64 bits, de los cuales 56 son usados para la encriptación, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso.
Como la clave efectiva es de 56 bits, son posible un total de 2^{56} = 72.057.594.037.927.936 claves posibles, es decir, unos 72.000 billones de claves, por lo que la ruptura del sistema por fuerza bruta o diccionario es súmamente improbable, aunque no imposible si se dispone de suerte y una gran potencia de cálculo.

3.1.3.2. Problemas al trabajar con DES:

PROBLEMAS GENERALES DEL ALGORITMO.

- ***Distribución de Claves:*** el emisor debe de comunicar su clave al receptor.
- ***Gestión de Claves:*** debe haber una clave por cada pareja de comunicantes.

INCONVENIENTES.

- Se considera un secreto nacional de EEUU, por lo que está protegido por leyes específicas, y no se puede comercializar ni en hardware ni en software fuera de ese país sin permiso específico del Departamento de Estado.

- La clave es corta, tanto que no asegura una fortaleza adecuada. Hasta ahora había resultado suficiente, y nunca había sido roto el sistema. Pero con la potencia de cálculo actual y venidera de los computadores, y con el trabajo en equipo por Internet se pudo violar este algorítmo, aunque eso sí, en un plazo de tiempo que no resultó peligroso para la información cifrada.

- No permite longitud de clave variable, con lo que sus posibilidades de configuración son muy limitadas, además de permitirse con ello la creación de limitaciones legales.

- La seguridad del sistema se ve reducida considerablemente si se conoce un número suficiente textos elegidos, ya que existe un sistema matemático, llamado Criptoanálisis Diferencial, que puede en ese caso romper el sistema en 2^{47} iteraciones.

3.1.3.3. Ventajas.

Entre las ventajas que ofrece **DES,** caben citar:

- ✓ Es el sistema más extendido del mundo, el que más máquinas usan, él más barato y el más probado.
- ✓ Es muy rápido y fácil de implementar.

Toda la seguridad de este sistema está basada en la llave simétrica, por lo que es misión fundamental tanto del emisor como del receptor conocer esta clave y mantenerla en secreto. Si la llave cae en manos de terceros, el sistema deja de ser seguro, por lo que habría que desechar dicha llave y generar una nueva.

Para que un algorítmo de este tipo sea considerado fiable debe cumplir varios requisitos básicos:
- Conocido el criptograma (texto cifrado) no se pueden obtener de él ni el texto en claro ni la clave.
- Conocidos el texto en claro y el texto cifrado debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.

Generalmente el algorítmo de encriptación es conocido, se divulga públicamente, por lo que la fortaleza del mismo dependerá de su complejidad interna y sobre todo de la longitud de la clave empleada, ya que una de las formas de

Criptoanálisis primario de cualquier tipo de sistema es la de prueba-ensayo, mediante la que se van probando diferentes claves hasta encontrar la correcta.

Los algorítmos simétricos encriptan bloques de texto del documento original, y son más sencillos que los sistemas de clave pública, por lo que sus procesos de encriptación y desencriptación son más rápidos.
Todos los sistemas criptográficos clásicos se pueden considerar simétricos, y los principales algorítmos simétricos actuales son **DES, TDES**, **IDEA** y **RC5**.

Hace algunos años atrás, EL NIST (National Institute of Standards Technology) convocó a un concurso para poder tener un sistema simétrico estándar, el cual se va a llamar **AES (*Advance Encryption Standard*).** Este algorítmo debe de reemplazar a **DES** en la mayor parte de las aplicaciones, debe ser seguro y que pueda usarse al menos en los próximos 20 años como estándar.

Las principales características que se pide a **AES** son que al menos sea tan seguro y rápido, es decir, que al menos evite los ataques conocidos. Una vez designado **AES** este podrá ser usado tanto como cifrador de bloques (Block Cipher), como cifrador de lluvia (Stream Cipher), como función resumen (hash function), y en el generador de números pseudoaleatorios.

La idea general de **AES,** es aplicar diferentes funciones al mensaje que se quiere cifrar de tal modo que solo conociendo una clave pueda aplicarse de forma inversa para poder así descifrar.

3.1.3.4. Modo de Operar de DES.

Dependiendo de la naturaleza de la aplicación **DES** tiene 4 modos de operación para poder implementarse:

- **ECB (Electronic Codebook Mode)**.

  Como Trabaja:
  - Libro Electrónico ECB.
  - Se cifran bloques de 64 bits.

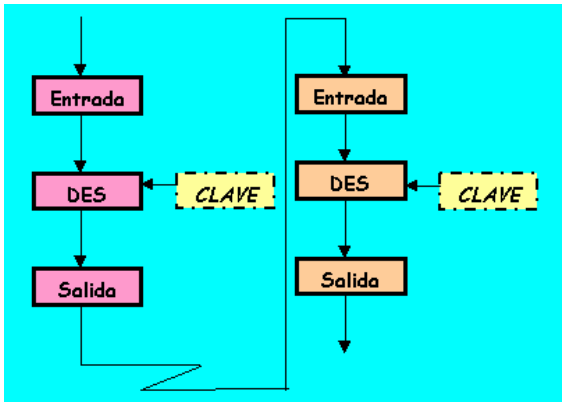


Figura 3.3

- Se utiliza para cifrar mensajes cortos de menos de 64 bits (un solo bloque de 64 bits) como claves. En otras palabras cifra cada bloque de 64 bits del mensaje en claro uno tras otro con la misma clave de 56 bits. Un par de bloques idénticos de mensaje en claro producen bloques idénticos de mensaje cifrado.
- Cada bloque cifrado es independiente de los demás.

Problema:

- Es posible cambiar el orden de bloques cifrados.
- Si el mensaje es repetitivo, puede ayudar al intruso a descifrarlo.

CBC (Cipher Block Chaining Mode)

Como Trabaja:

- Encadenamiento de Bloques.
- Para mensajes largos.
- Se codifica [$Bloque_{64} \oplus Salida\_Bloque\_Anterior_{64}$].

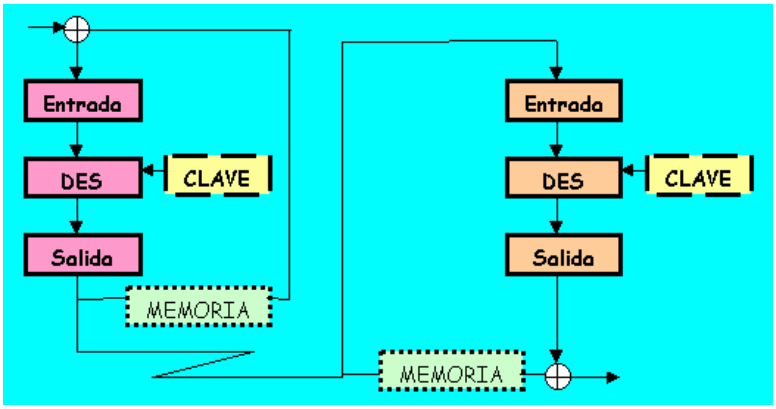


Figura 3.4

- Se utiliza en codificación de mensajes largos o en bloques más de 64 bits.
- Sobre cada bloque de 64 bits del mensaje en claro se ejecuta un OR exclusivo con el bloque previo del mensaje cifrado antes de proceder al cifrado con la clave **DES**. De este modo, el cifrado de cada bloque depende del anterior y bloques idénticos de mensaje en claro producen diferentes mensajes cifrados.

Problema:

- Propaga errores de transmisión al siguiente bloque.
- Requiere bloque de inicialización.
- Se requiere tamaños de bloques de 64 bits para codificar.

CFB (Cipher Block Feedback).

Como Trabaja:

- Modo de Realimentación cifrada.
- Se utiliza para codificar información que se presenta en bloques de 8 bits.

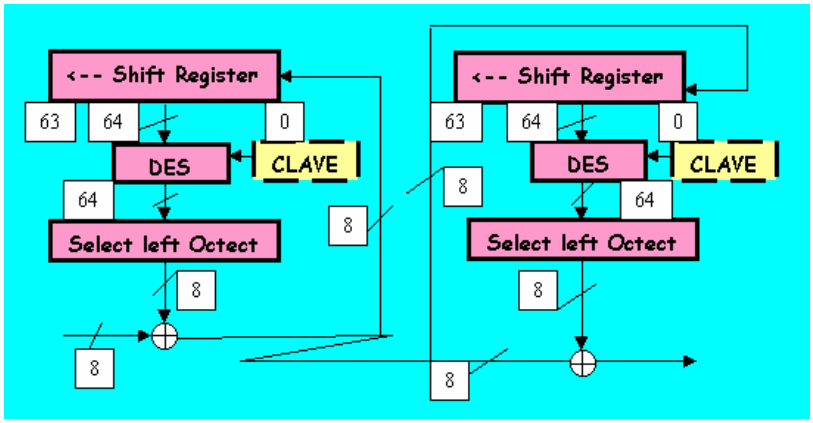


Figura 3.5

- Para cifrar bit por bit o byte por byte.
- El cifrado de un bloque de mensaje en claro procede de ejecutar un OR exclusivo del bloque de mensaje en claro con el bloque previo cifrado. CFB puede modificarse para trabajar con bloques de longitud inferior a 64 bits.

Problema:

- Propaga errores de transmisión durante las siguientes 8 decodificaciones.
- Más lento.
- Requiere bloque de inicialización.

OFB (Output FeedBack Block).

Como Trabaja:

- Método de Realimentación de Salida.
- Se requieren Bloques de "m" bits.

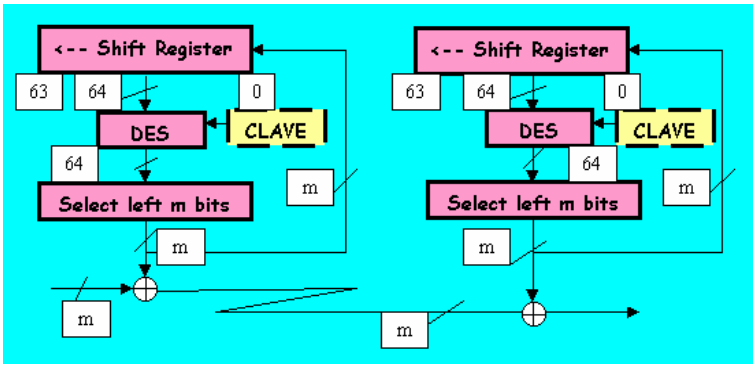


Figura 3.6

- No se propagan los errores de Transmisión.
- Similar al modo CFB excepto en que los datos sobre los que se ejecuta el OR exclusivo junto con los bloques de mensaje en claro es generada independientemente del mensaje en claro y del mensaje cifrado.

Problema:

- Necesita Bloque de inicialización.
- Se puede producir pérdida de sincronísmo.
- Menos seguro: Posible periodicidad del mensaje puede ayudar a descifrar.

3.1.3.5. Seguridad de DES.

DES puede ser atacado mediante la fuerza bruta, probando todas las claves posibles (2^{56}), siendo este algorítmo de una complejidad O(2^{55}). A pesar de los rumores que aseguraban que el NBS modificó el algorítmo para hacerlo más débil, aún no ha sido roto públicamente más que por el ataque a fuerza bruta. Aunque ha habido diferentes análisis que han demostrado que puede disminuirse la complejidad del problema hasta 2^{43}, han resultado de implementación poco práctica.

Sin embargo, la sensación de que **DES** tiene sus días contados se ha ido avivando durante los últimos años. El 18 de junio de 1997, un esfuerzo coordinado a través de Internet, en respuesta a un desafío de *RSA Data Security, Inc.,* ha permitido el descifrado de un mensaje cifrado con una clave **DES** de 56 bits utilizando la fuerza bruta.
El tiempo invertido desde que comenzó el análisis de las posibles claves hasta que se consiguió descifrar el mensaje fue de cuatro meses.

Tras el lanzamiento del segundo desafío (el 13 de julio de 1998) y, con el fin de demostrar la inseguridad de **DES**, la *Electronic Frontier Foundation* anunció la construcción de una máquina especializada diseñada para reventar mensajes cifrados con **DES**. Esta máquina, bautizada apropiadamente como **DES Cracker**, fue construida por menos de 250.000 dólares, y fue capaz de ganar fácilmente el segundo desafío, tardando apenas tres días. La exportación de la propia máquina no está permitida por las leyes de EE.UU.

Actualmente **DES** ya no es un estándar y fue roto en Enero de 1999 con un poder de cómputo (método a fuerza bruta, es decir, probando todas las 2^{56} posibles claves), que efectuaba aproximadamente 250 mil millones de ensayos en un segundo.
Lo anterior quiere decir que, es posible verificar todas las claves posibles en el sistema **DES** en un tiempo corto, lo que lo hace inseguro para propósitos de alta seguridad. La opción que se ha tomado para poder suplantar a **DES** ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algorítmo par fortalecer la longitud de la clave, esto a tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como **TRIPLE - DES** o **TDES**.

3.1.4. ALGORÍTMO TDES (*TRIPLE DES*).

Como hemos visto, el sistema o algorítmo **DES** se considera en la actualidad poco práctico, debido a la corta longitud de su clave. Para solventar este problema y

continuar utilizando **DES** se hizo una mejora y se creó el sistema **Triple DES** o **TDES**, basado en tres iteraciones sucesivas del algorítmo **DES** (el mensaje es cifrado tres veces), con lo que se consigue una longitud de clave de 192 bits, y que es compatible con **DES** simple.

Existen varias implementaciones de **TDES**:

1. **DES-EEE3**: Se cifra tres veces con una clave diferente cada vez.
2. **DES-EDE3**: Primero se cifra, luego se descifra y por último se vuelve a cifrar, cada vez con una clave diferente.
3. **DES-EEE2** y **DES-EDE2**: Similares a los anteriores con la salvedad de que la clave usada en el primer y en el último paso coinciden.

Se estima que las dos primeras implementaciones, con claves diferentes, son las más seguras. Si se quiere romper el algorítmo usando la fuerza bruta, la complejidad asciende a $O(2^{112})$.

3.1.4.1. Descripción del Algorítmo.

Para implementarlo, se toma una clave de 128 bits y se divide en 2 diferentes de 64 bits (c1=c2), para que el sistema se comporte como **DES** simple.

De manera general **TDES** funciona así, aplicándose el siguiente proceso al documento en claro:

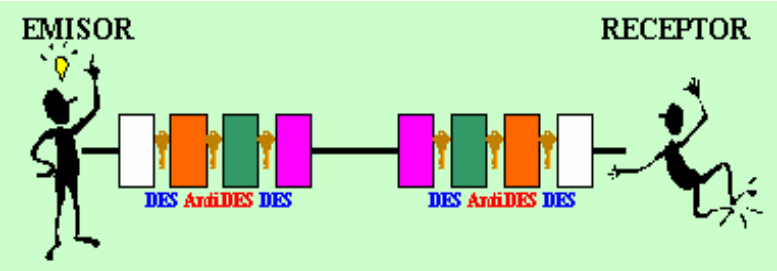


Figura 3.7

1. Se le aplica al documento a cifrar un primer cifrado mediante la primera clave, C1.
2. Al resultado (denominado AntiDES) se le aplica un segundo cifrado con la segunda clave, C2.
3. Y al resultado se le vuelve a aplicar un tercer cifrado con la primera clave, C1.

Tras un proceso inicial de búsqueda de compatibilidad con **DES**, 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.

TDES, usa entonces una clave de 168 bits, aunque se ha podido mostrar que los ataques actualmente pueden romper **TDES** con una complejidad de 2^{112} operaciones para obtener la clave a fuerza bruta, además de la memoria requerida.

Se optó por **TDES** ya que es muy fácil interoperar con **DES** y proporciona seguridad a mediano plazo.

3.1.4.2. Problema al trabajar con TDES.

El problema principal que presenta el algorítmo **TDES,** es básicamente la necesidad de gestionar dos claves.

3.1.4.3. Ventajas.

Una de las grandes ventajas que presenta el uso de **TDES** es que no es necesario una computadora muy cara ni mucho tiempo de procesamiento para conseguir encontrar la clave.

Además, mejora las técnicas criptoanalístas: fuerza bruta, diferencial.
Mejora la potencia de los sistemas informáticos: electrónica, sistemas distribuidos.

3.1.4.4. Modo de Operar de TDES.

TDES, consiste en aplicar 3 veces **DES** de la siguiente manera:

- ***La primera vez***, se usa una clave **K1** junto con el bloque **BO**, de forma ordinaria **E** (de Encriptar), obteniendo el bloque **B1**.

- ***La segunda vez,*** se toma a **B1** con la clave **K2**, diferente a **K1** de forma inversa, llamada **D** (Desencriptar).

- ***La tercera vez,*** a **B2** con una clave de **K3** diferente a **K1** y **K2**, de forma ordinaria **E** (Encriptar), es decir, aplica de la interacción 1 a la 16 al **BO** con la clave **K1**, después aplica de la 16 a la 1, a **B1** con la clave **K2**, finalmente aplica una vez mas de la 1 a la 16 a **B3** usando la clave **K3**, obteniendo finalmente a **B3**.

Sintetizando lo anteriormente expuesto **TDES,** utiliza dos claves **K1 y K2** (con el objeto de aumentar la longitud de las pruebas), en tres etapas:

- **Cifrado: E (K1).**
- **Descifrado: D (K2).**
- **Cifrado: E (K1).**

Si **K1** = **K2,** son compatibles con los sistemas clásicos **DES;** aumentando así el tiempo de procesamiento para averiguar la clave exponencialmente **2^{142} =5,2 x 10^{13}.**

En cada una de estas tres veces aplica el medio de operación más adecuado.

El proceso del cifrado con **TDES** se puede apreciar en la siguiente figura:

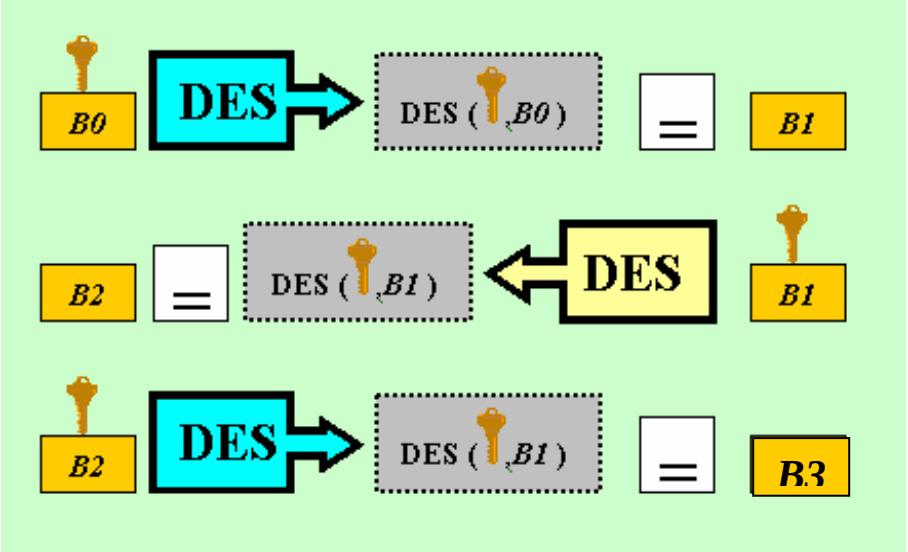


Figura 3.8

Este hecho se basa en que **DES** tiene la característica matemática de no ser un grupo, lo que implica que si encripta el mismo bloque dos veces con dos llaves diferentes se aumenta el tamaño efectivo de la llave.

En los últimos 20 años se han diseñado una gran cantidad de sistemas criptográficos simétricos, entre algunos de ellos están: **RC- 4, IDEA, FEAL, LOKI´ 9, DESX, Blowfish, CAST, GOST.**

Sin embargo no han tenido el alcance de **DES**, a pesar de que algunos de ellos tienen mejores propiedades.

3.1.5. ALGORÍTMO IDEA (*INTERNATIONAL DATA ENCRYPTION ALGORITHM*).

Debido a la previsible retirada de **DES**, prestigiosos criptógrafos de todo el mundo han estado trabajando desde los últimos años de la década de los ochenta para encontrar algoritmos, de un lado compatibles con **DES** (debido al gran número de productos basados en **DES** utilizados en todo el mundo), y de otro, lo suficientemente robustos como para sustituirle con garantías.

3.1.5.1. Descripción del Algoritmo.

IDEA, fue desarrollado en Suiza por Xuejia Lai y James Massey en 1992. Se trata de un algorítmo iterativo que trabaja sobre bloques de 64 bits, operando siempre con números de 16 bits y utilizando claves de 128 bits.
El proceso de cifrado consta de ocho fases idénticas en la que lo único que varía es el sub-bloque de clave utilizada, terminando el cifrado con una transformación de la salida. En cada paso se utilizan tres operaciones: suma bit a bit, multiplicación bit a bit y OR exclusivo.

El algoritmo de desencriptación es muy parecido al de encriptación, por lo que resulta muy fácil y rápido de programar, y hasta ahora no ha sido roto nunca, aportando su longitud de clave una seguridad fuerte ante los ataques por fuerza bruta (prueba y ensayo o diccionarios).
Este algorítmo es de libre difusión y no está sometido a ningún tipo de restricciones o permisos nacionales, por lo que se ha difundido ampliamente, utilizándose en sistemas como UNIX y en programas de cifrado de correo como PGP.

3.1.5.2. Modo de Operar de IDEA.

- Se codifican bloques de 64 bits que se dividen en 4 bloques de 16 bits.
- 8 etapas de operaciones y una transformación final.
- Operaciones con números de 16 bits:
  - Sumas módulo 2^{16} de 16 bits.
  - Multiplicaciones módulo 2^{16} de 16 bits.
  - OR EXCLUSIVO de 16 bits.
- Con la clave (128) se generan 52 subclaves (16) usadas en las 8 etapas y en la transformación.

3.1.5.3. Seguridad de IDEA.

Se estima que **IDEA** es más seguro que **DES**. Las bases teóricas y la implementación de los diferentes sistemas criptográficos que lo utilizan están sometidos a público escrutinio. Los programas actuales que implementan **IDEA** son tan rápidos como los que implementan **DES**.
Se puede utilizar con los métodos descritos en DES: CBC, CFB.

3.1.6. OTROS ALGORÍTMOS.

Entre otros Algoritmos que dan soporte a la ***Criptografía Simétrica*** caben señalar:

3.1.6.1. RC-2 y RC-4.

RC-2 y **RC-4** fueron desarrollados por Ron Rivest, uno de los coautores del algorítmo RSA (como veremos mas adelante), en 1989 y 1987, respectivamente.

Durante varios años, se ha tratado de algorítmos con propietario y sus detalles no fueron hechos públicos. De este modo, su seguridad se basaba en el prestigio de su autor y en el respaldo que les daba *RSA Data Security, Inc.* para las que los desarrolló. Sin embargo, en 1994 **RC4** fue sometido a Ingeniería Inversa y los resultados publicados en Internet. Las pruebas hechas a dicho diseño se ajustan a los resultados esperados por lo que es razonable pensar que el diseño publicado es correcto. **RC2** fue finalmente publicado como ***Internet Draft*** en 1997.

RC-2 es un cifrador de bloque con una longitud de clave variable. Tiene definido los mismos modos que **DES** y, con una clave de 64 bits, su implementación en *software* es dos o tres veces más rápida que la de **DES**.

RC-4, al igual que **RC-2**, tiene una longitud de clave variable. Sin embargo, se trata de un cifrador de flujo.

Como el resto de productos criptográficos desarrollados en EE.UU., las restricciones a su exportación son muy fuertes. **RC-2** y **RC-4** tienen un status especial que facilita la concesión de licencias de exportación de productos basados en ellos, pero limitando la longitud de su clave a 40 bits. **RC-4** y **RC2** se usan en centenares de productos comerciales, al igual que lo hace el protocolo **SSL**

Entre otros algorítmos basados en clave simétrica son **SkipJack**, que utiliza claves de 80 bits para cifrar mensajes en bloques de 64 bits (sólo implementado en *hardware*) y que constituye la base del controvertido chip *Clipper*; **FEAL**, **SAFER** o **Blowfish** (diseñado por Bruce Schneier).

.

3.1.7. RESUMEN.

Para sintetizar la explicación y entender el funcionamiento básico de un ***Criptosistema Simétrico*** podemos representarlo gráficamente con los siguientes pasos:

CLAVE ÚNICA:

En el envío de un mensaje debe aparecer un **Emisor** y un **Receptor**, ambos deben poseer la misma **Clave Privada** para poder descifrar el mensaje enviado de uno a otro.

Los pasos a realizar son los siguientes:

EMISOR:

1. El **Emisor** escribe el mensaje origen con destino al **Receptor**.

Figura 3.9

2. Luego cifra (encripta) el mensaje con su clave privada.

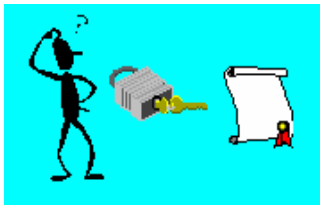


Figura 3.10

3. Se da el envío del mensaje al destino.

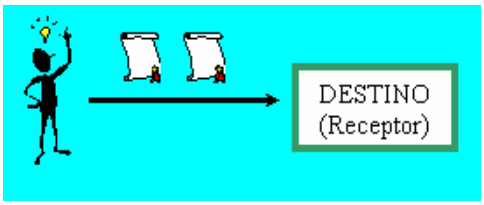


Figura 3.11

RECEPTOR:

1. El **Receptor** recibe el mensaje a través de la red.

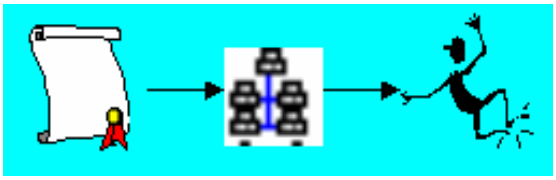


Figura 3.12

.

2. Luego descifra (descripta) con su clave privada el mensaje cifrado por el **Emisor**.

Figura 3.13

3. Obtiene el mensaje original que le envío el **Emisor**.

Figura 3.14

Conclusiones:

1. El Emisor y el Receptor son los únicos conocedores de la clave privada.
2. Se necesita tener un número muy alto de claves secretas, en concreto, una para cada persona con la que nos comuniquemos.
3. Para Desencriptar es necesario conocer la clave de encriptación.

Entre los ataques más potentes a la ***Criptografia Simétrica*** están: Criptoanálisis Diferencial y Lineal.

Sin embargo no han podido ser muy eficientes en la práctica por lo tanto, por el momento después de que un sistema criptográfico es publicado y se muestra inmune a estos dos tipos de ataques. La mayor preocupación es la longitud de las claves.

El problema de la ***Criptografía de Llave Privada***, es que en una red muy grande, en caso de que se decida cambiar la clave de desciframiento, hay que notificar a

cada uno de los participantes en los procesos de transmisión de datos, corriéndose el peligro de que caiga la nueva clave en manos no autorizadas.

Otro de los problemas que enfrenta la ***Criptografia Simétrica***, es la distribución de las claves, el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.
Cuando se utiliza únicamente ***Criptografía de Clave Simétrica***, aunque el sistema de generación de claves suele ser sencillo, ya que no se requiere una gran infraestructura para soportarlo, los mecanismos de distribución de las claves suelen ser muy complejos.

En este caso, los principales parámetros que hay que tener en cuenta son el modo de difundir la clave secreta de forma segura a las dos entidades que van a utilizarla y la frecuencia con la que se deben renovar las claves para evitar que sean desveladas.

3.2. CRIPTOGRAFÍA ASIMÉTRICA O DE CLAVE PÚBLICA.

3.2.1. INTRODUCCIÓN.

El nacimiento de la ***Criptografía Asimétrica*** se dió al estar buscando un modo mas práctico de intercambiar las llaves simétricas. Diffie y Hellman, proponen una forma para hacer esto, sin embargo no fue hasta que el popular método de Rivest Shamir y Adleman **RSA** publicado en 1978, cuando toma forma la ***Criptografía Asimétrica.***

La ***Criptografía Asimétrica*** o ***de Clave Pública***, se basa en la imposibilidad computacional de factorizar números enteros grandes usando dos claves diferentes, claves que poseen una propiedad fundamental: una clave puede desencriptar lo que la otra ha encriptado.

La ***Criptografía Asimétrica*** es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar, denominada **Clave Pública** (conocida por todos), es usada por el propietario para encriptar los mensajes, mientras que la otra, llamada **Clave Privada** (conocida solo por el propietario), es usada para desencriptar o descifrar el mensaje cifrado.
La asimetría se refiere a que si codificamos con la clave publica solo se puede decodificar con la clave privada.

En este sistema, para enviar un documento con seguridad, el emisor encripta el mismo con la clave pública del receptor y lo envía por el medio inseguro. Este documento está totalmente protegido en su viaje, ya que sólo se puede desencriptar con la clave privada correspondiente, conocida sólamente por el receptor. Al llegar el mensaje cifrado a su destino, el receptor usa su clave privada para obtener el mensaje en claro.

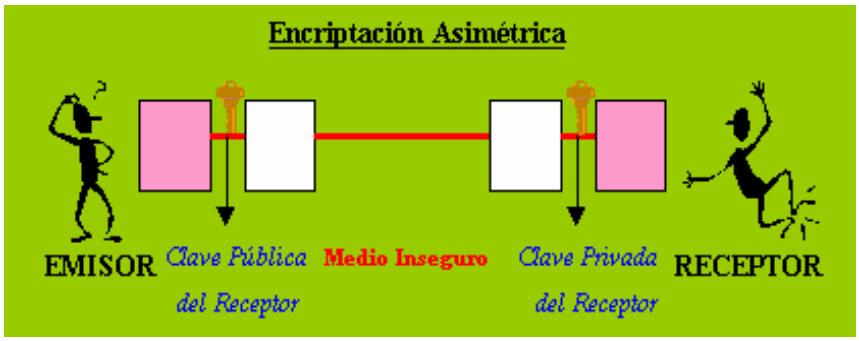


Figura 3.15

Las claves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves públicas son diferentes, entonces sus llaves privadas asociadas también lo son, y viceversa.

Los algorítmos asimétricos están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso, salvo que se conozca la clave privada, como la potencia y el logarítmo. Ambas claves, pública y privada, están relacionadas matemáticamente, pero ésta relación debe ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la otra.
Este es el motivo por el que normalmente estas claves no las elige el usuario, si no que lo hace un algorítmo específico para ello, y suelen ser de gran longitud.

La clave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la clave pública es difundida ampliamente por Internet, para que esté al alcance del mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas claves.

3.2.2. CLASIFICACIÓN.

En la actualidad la ***Criptografía Asimétrica*** o de **Clave Pública** se divide en tres familias, según el problema matemático del cual basan su seguridad:

- **Primera Familia:** La que basa su seguridad en el problema de Factorización Entera **PFE**, los sistemas que pertenecen a esta familia son, el sistema **RSA,** y el Rabin Williams **RW.**
- **Segunda Familia:** Es la que basa su seguridad en el problema del Logarítmo Discreto **PLD,** a esta familia pertenece el sistema de Diffie Hellman **DH** de intercambio de claves y el sistema **DSA** de Firma Digital.
- **Tercera Familia:** es la que basa su seguridad en el problema del Logarítmo Discreto Elíptico **PLDE,** en este caso hay varios esquemas tanto de intercambio de claves como de Firma Digital que existen como **DHE (**Diffie Hellman Elíptico**), DSAE (**Nyberg-Rueppel**), NRE (**Menezes, Qu, Vanstone**), MQV.**

Aunque a las familias anteriores pertenecen los sistemas asimétricos más conocidos, existen otros tipos que basan su seguridad en otro tipo de problema como por ejemplo en el problema del **Logaritmo Discreto Hiperhelíptico**, sobre problemas de retículas y sobre subconjunto de clases de campos numéricos reales y complejos.

Actualmente la ***Criptografía Asimétrica*** es muy usada, sus dos principales aplicaciones son precisamente: **El Intercambio de Claves Privadas y la Firma Digital**, de la cual se habló anteriormente.

3.2.3. ALGORÍTMO RSA *(RIVEST SHAMIR ADLEMAN)*.

El algorítmo de clave pública **RSA**, fué creado en 1978 por Rivest, Shamir y Adleman, y es el sistema criptográfico asimétrico más conocido y usado. Estos señores se basaron en el artículo de Whitfield Diffie y Martín Hellman (Diffie - Hellman), sobre sistemas de llave pública, crearon su algorítmo y fundaron la empresa **RSA Data Security Inc.**, la cual es actualmente una de las más prestigiosas en el entorno de la protección de datos.

RSA es el más conocido y usado de los sistemas de clave pública, y también el más rápido de ellos. Presenta todas las ventajas de los sistemas asimétricos, incluyendo la Firma Digital, aunque resulta más útil a la hora de implementar la confidencialidad el uso de sistemas simétricos, por ser más rápidos. Se suele usar también en los sistemas mixtos para encriptar y enviar la clave simétrica que se usará posteriormente en la comunicación cifrada.

El sistema **RSA** se basa en el hecho matemático de la dificultad de factorizar números muy grandes. Para factorizar un número el sistema más lógico consiste en empezar a dividir sucesivamente éste entre 2, entre 3, entre 4,..., y así sucesivamente, buscando que el resultado de la división sea exacto, es decir, de resto 0, con lo que ya tendremos un divisor del número.

3.2.3.1. Descripción de RSA.

Si el número considerado es un número primo (el que sólo es divisible por 1 y por él mismo), tendremos que para factorizarlo habría que empezar por 1, 2, 3,...., hasta llegar a él mismo, ya que por ser primo ninguno de los números anteriores es divisor suyo. Y si el número primo es lo suficientemente grande, el proceso de factorización es complicado y lleva mucho tiempo.
Basado en la exponenciación modular de exponente y módulo fijos, el sistema **RSA** crea sus claves de la siguiente forma:

1. Se buscan dos números primos lo suficientemente grandes: **p** y **q** (de entre 100 y 300 dígitos).
2. Se obtienen los números **n** = p * q y **Ø** = (p-1) * (q-1).
3. Se busca un número **e** tal que no tenga múltiplos comunes con **Ø**.
4. Se calcula **d** = e^{-1} mod **Ø**, con mod = resto de la división de números enteros.

Y ya con estos números obtenidos, **n es la clave pública y d es la clave privada**. Los números **p, q y Ø** se destruyen. También se hace público el número **e,** necesario para alimentar el algorítmo.
El cálculo de estas claves se realiza en secreto en la máquina en la que se va a guardar la clave privada, y una vez generada ésta conviene protegerla mediante un algorítmo criptográfico simétrico.
En cuanto a las longitudes de claves, el sistema **RSA** permite longitudes variables, siendo aconsejable actualmente el uso de claves de no menos de 1024 bits (se

han roto claves de hasta 512 bits, aunque se necesitaron más de 5 meses y casi 300 ordenadores trabajando juntos para hacerlo).

Método para enviar la información encriptada:
- Se cifra con la clave publica del receptor.
- Solo el receptor puede descifrar con su clave privada.

Cada usuario sólo necesita tener un par de claves, la privada y la pública.
Todos tenemos acceso a la clave pública de nuestro interlocutor.
No debe ser factible averiguar la clave privada de la clave pública.

Los algorítmos se basan en la complejidad computacional asociada a ciertas operaciones:
- Mochila: averiguar un conjunto de valores conociendo la suma de sus pesos.
- Factorización de números grandes:
- Logarítmos Discretos.
- Curvas Elípticas.

Se buscan funciones trampa.

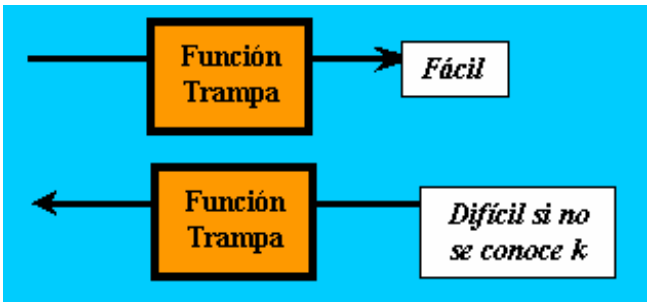


Figura 3.16

Ejemplo $y = a^x$
- Es fácil calcular **y** conocidos **a** y **x**.
- Difícil calcular **x** conocidos **a** e **y**, o sea realizar **$x = \log_a y$**.

Algunos algorítmos basados en clave publica: **RSA, ElGamal, DSA, Knapsack.**

3.2.3.2. Funcionamiento de RSA.

En el caso de **RSA**, el problema matemático es el de la Factorización de un número entero **n** grande (1024 bits), este número entero se sabe es producto de dos números primos **p, q** de la misma longitud, entonces la clave pública es el número **n** y la privada es **p, q**. Cada una de estas dos diferentes aplicaciones consiste en una serie de pasos que a continuación se describen.

a. A cada usuario se le asigna un numero entero **n**, que funciona como su clave pública.

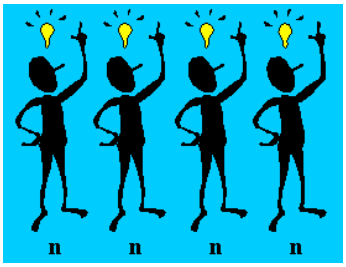


Figura 3.17

b. Solo el usuario respectivo conoce la Factorización de **n** (o sea p,q), que mantiene en secreto y es la clave privada.

Figura 3.18

c. Existe un directorio de claves públicas.

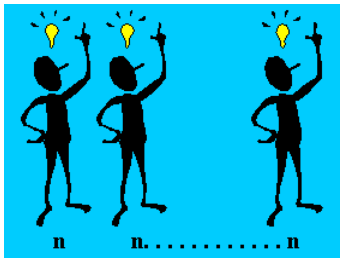


Figura 3.19

d. Si alguien quiere mandar un mensaje **m** a algún usuario entonces elige su clave pública **n,** y con información adicional también pública, puede mandar el mensaje cifrado **c**, que solo podrá descifrar el usuario correspondiente, el mensaje **m** convertido a número (codificación), se somete a la siguiente operación:

$$c = m^e \bmod n$$

Figura 3.20

e. Entonces el mensaje **c** puede viajar sin problemas por cualquier canal inseguro.

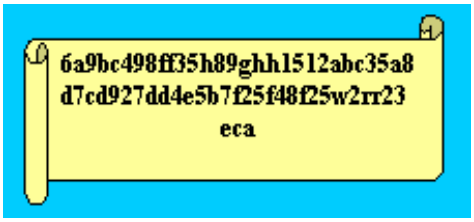


Figura 3.21

f. Cuando la información cifrada llega a su destino, el receptor procede a descifrar el mensaje con la siguiente fórmula:

$$m = c^d \bmod n$$

Figura 3.22

g. Se puede mostrar que estas fórmulas son inversas y por lo tanto dan el resultado deseado, **(m,e)** son públicos y se pueden considerar como la clave pública, la clave privada es la pareja **(p,q)** o equivalente el numero **d**. La relación que existe entre **d** y **e** es que uno es el inverso multiplicativo del otro modulo **λ(n);** donde **λ(n)** es el mínimo común múltiplo de **p-1** y **q-1**, esto significa que la clave privada o la pareja p,q es el número d.

Ejemplo:

- Supongamos p=47 y q=59
- Por tanto, n=p*q=2773
- De estos datos, se calcula (p-1)(q-1)=2668
- Eligiendo e=17, calculamos d utilizando algorítmos de factorización (d=157)

Por tanto, la clave pública P será el par (17, 2773), mientras que la privada, S, la constituirá el par (157, 2773).

Si aumentamos la longitud de la clave aumenta la complejidad para el criptoanalista, pero también se hace más lento su uso (actualmente 1024 bits de la clave se considera seguro).

Se opera con las dos claves indistintamente:

- Cifra con clave pública (todos) y descifrar con clave privada (solo propietario).
- Cifra con clave privada (solo propietarios) y descifrar con clave pública (todos).

- **Cifrado con Clave Privada:**
  No Confidencialidad: cualquiera puede descifrar.
  Autenticidad: garantía de que ha cifrado el propietario de la clave privada.
- **Cifrado con Clave Pública:**
  Confidencialidad: solo el propietario de la clave privada puede descifrar.

En términos muy generales es así como funciona el sistema **RSA.** Sin embargo en la realidad existen dos formas que son las mas comunes; estas formas depende de la aplicación y se llaman **Esquema de Firma** y el **Esquema de Cifrado.**

3.2.3.3. Rapidez de RSA.

Fruto de los requerimientos para hacer seguro **RSA**, surge su principal problema: su lentitud. En general, se elige como clave pública el menor de los dos exponentes a fin de conseguir que el proceso de cifrado sea el más rápido (más que el descifrado, el cual, a su vez lo es más que la generación de claves). El cifrado, que utiliza la clave pública, tiene una complejidad de $\mathbf{O(k^2)}$, el descifrado $\mathbf{O(k^3)}$ y la generación de claves $\mathbf{O(k^4)}$, en donde **k** es la longitud en bits del módulo.
Una práctica habitual es elegir como clave pública un exponente pequeño, típicamente 1001 ó 10001, variando el módulo.

3.2.3.4. Seguridad de RSA.

Si un intruso que quisiera quebrar el algorítmo fuese capaz de factorizar **n** (parte de la clave pública), entonces podría utilizar estos factores para deducir rápidamente **e** y **d**. Por lo tanto, si fuese fácil factorizar números grandes, sería fácil romper **RSA**. Lo contrario, es decir, si por ser difícil factorizar números muy grandes es difícil romper **RSA** no se ha demostrado (de hecho, tampoco se ha demostrado que factorizar números muy grandes sea en realidad un problema difícil).
Para lograr la máxima seguridad, es necesario utilizar enteros de más de 100 dígitos de longitud, pues factorizar números más pequeños es posible. Según asegura Bruce Schneier en *Applied Cryptography*, en 1996, un número de 129 dígitos decimales está en el borde de la tecnología factorizadora. Además, se debe asegurar que el producto (p-1)(q-1) no tiene factores primos pequeños.

Además, **RSA** basa su seguridad en el problema de la Factorización de un Numero ***n*** Entero Grande, es una función computacionalmente segura, ya que si bien realizar la exponenciación modular es fácil, su operación inversa, la extracción de raíces de módulo ***Ø*** no es factible a menos que se conozca la factorización de **e**, clave privada del sistema.

3.2.3.5. Problemas al trabajar con RSA.

Cuando se utiliza Criptografía de Clave Pública, el sistema de gestión de claves se complica:

- En primer lugar es necesario almacenar las claves públicas en un lugar al que tengan libre acceso todos los usuarios que forman parte del entorno de seguridad. ITU, en su recomendación X.509, propone la utilización del Directorio para este fin; pero no todos los usuarios de seguridad tienen acceso

al Directorio X.500, por lo que en muchos entornos es necesario crear o utilizar otro tipo de bases de datos.

- En segundo lugar, la lentitud es un problema que afecta en gran parte a **RSA,** esto se debe al aumento de longitud de la clave**.**

3.2.3.6. Cuadro Resumen.

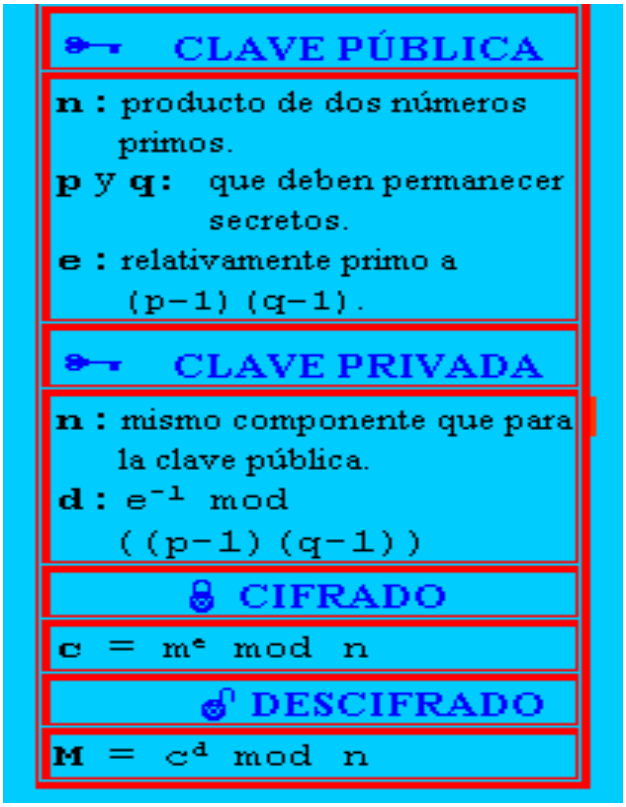


Figura 3.23

3.2.4. ALGORITMO CCE *(CRIPTOGRAFÍA CON CURVAS ELÍPTICAS)*.

3.2.4.1. Introducción.

Otro tipo de Criptografía de Clave Pública es el que usa ***Curvas Elípticas*** definidas en un campo finito. La diferencia que existe entre este sistema y **RSA,** es el problema del cual basan su seguridad, mientras **RSA** razona de la siguiente manera: *Te doy el número 15 y te reto a encontrar los factores primos*. El problema del cual están basados los sistemas que usan Curvas Elípticas que denotaremos como **CCE,** es el problema del logarítmo discreto elíptico, en este caso su razonamiento con números sería algo como: *Te doy el número 15 y el 3 y te reta a encontrar cuantas veces tienes que sumar el mismo 3 para obtener 15.*

En lo que sigue nos dedicaremos a explicar un poco de lo más importante de **CCE:**

***¿QUÉ ES UNA CURVA ELIPTICA*?**

Es un conjunto finito de puntos **P, Q...,S** donde cada punto es una pareja **P= (x,y)** y las coordenadas **x,y** satisfacen una ecuación de la siguiente forma:

$$y^2 + axy + by = x^3 + cx^2 + dx + e.$$

Ecuación 1.

Donde las constantes **a, b, c, d** y **e** pertenecen a cierto conjunto llamado campo **F,** que para propósitos de la ***Criptografía Moderna*** o es un campo primo **(Z_p)** o un campo de características 2, o sea donde los elementos son n-adas de ceros y unos **(F_2^n)**.

El conjunto de puntos que satisfacen a una ecuación similar a la *Ecuación 2,* lo podemos representar como: **E: $O,P_1.P_2,P_3,...P_n$.**

Este conjunto de puntos pueden sumarse y tiene las mismas propiedades que la suma de los números enteros, es decir lo que se conoce como un grupo abeliano, hay que hacer notar que en este caso el que hace el papel de cero (identidad aditiva) es un punto especial que no tiene coordenadas y se representa como **O** llamado punto al infinito.

La suma de estos puntos tiene una explicación geométrica muy simple, en este caso la gráfica representa a todos los puntos que satisfacen la *Ecuación 1*.
Si suponemos que queremos sumar a **P** y **Q**, trazamos una línea recta que pase por **P** y **Q**, la *Ecuación 3*, es de grado 3 y la línea de grado 1, entonces existe siempre tres soluciones, en este caso la tercera solución esta dibujada como el punto **–P-Q.** Enseguida se procede a dibujar una línea recta paralela al eje Y que pase por **–P-Q**, esta línea vertical también interceptan al punto especial llamado infinito y que geométricamente esta en el horizonte del plano, el tercer punto es por definición **P+Q**, como se muestra en la figura:

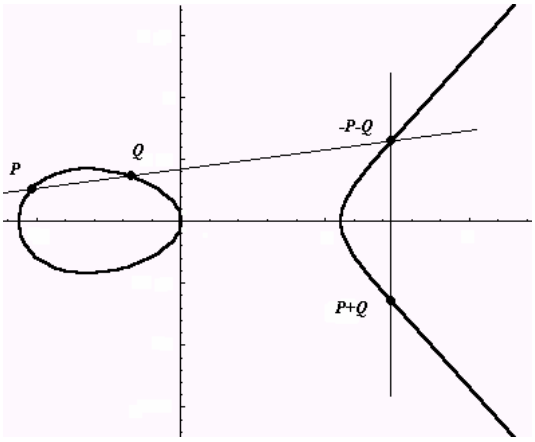


Figura 3.24

La anterior forma de sumar puntos de una curva elíptica es un poco extraña sin embargo, ésta extrañeza es lo que permite que sea un poco más difícil romper los **CCE.** En el área de las matemáticas conocida como **"Teoría de Grupos"** se sabe que estos grupos son muy simples llamados grupo finitos abelianos lo que permite también que los **CCE** sean fácil de implementar, llamaremos al número de puntos racionales de la curva como el orden de la curva.

3.2.4.2. Seguridad de CCE.

Los **CCE** basan su seguridad en el Problema del Logarítmo Discreto Elíptico (**PLDE**), esto quiere decir que dados **P,Q** puntos de la curva hay que encontrar un número entero **x** tal que **Xp = Q (Xp = P+P+...+ P, x veces)**.
Obsérvese que a diferencia del **PPE** (Problema de Factorización Entera), el **PLDE** no maneja completamente números, los que hace más complicado su solución.

La creación de un protocolo con ***Criptografía de Curvas Elípticas*** requiere fundamentalmente una alta seguridad y una buena implementación, para el primer punto se requiere que la elección de la curva sea adecuada, principalmente que sea no-supersingular y que el orden del grupo de puntos racionales tenga un factor primo de al menos 160 bits, además de que este orden no divida el orden de un número adecuado de extensiones del campo finito, para que no pueda ser sumergido en él, si el campo es Z_p, se pide que la curva no sea anómala. Todo esto con el fin de evitar los ataques conocidos.

Para el caso de la implementación hay que contar con buenos programas que realicen la aritmética del campo finito, además de buenos algorítmos que sumen puntos racionales, si el campo es **Z_p** existen varios y si el campo es **F_2^n** entonces se toma una base polinomial que tenga el mínimo de términos.
Ej: Un trinomio para generar los elementos del campo finito esto si la implementación es en software y se toma una base normal si es en hardware. Además de contemplar que las operaciones de puntos racionales pueden hacerse en el espacio proyectivo esto elimina el hacer divisiones ahorrando tiempo.

3.2.4.3. Ventajas.

Lo anterior se ve reflejado en las ventajas que ofrecen los **CCE** en comparación con **RSA,** la principal es la longitud de la clave secreta. Se puede mostrar que mientras **RSA** se tiene que usar una clave 1024 bits para ofrecer una considerable seguridad, los **CCE** solo usan 163 bits para ofrecer una considerable seguridad, así también las claves **RSA** de 2048 son equivalentes en seguridad a 210 de **CCE.** Esto se debe a que para resolver el **PLDE** el único algorítmo conocido toma tiempo de ejecución totalmente exponencial, mientras que el algoritmo que resuelve **PPE** incluso también el **PDL** en **Z_p** toman tiempo subexponencial.

Otra buena noticia sobre los **CCE** es que los elementos de los puntos racionales pueden ser elementos de un campo finito de característica 2, es decir pueden ser arreglos de ceros y unos de longitud finita (01001101110010010111), en este caso es posible construir una aritmética, a esto se le conoce como Base Normal Optima.

Lo anterior permite con mucho que los **CCE** sean idóneos para ser implementados en donde el poder de cómputo y el espacio del circuito sea reducido, donde será requerida una alta velocidad de procesamiento o grandes volúmenes de transacciones, desde el espacio de almacenamiento, la memoria o el ancho de banda sea limitado. Lo que permite su uso en Smart Cards, Teléfonos celulares, Fax, PC's.

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 Electronic Engineers).
- **ANSI X9,62, ANSI X9,63, ANSI TG-17, ANSI X12.** (American National Standards Institute).
- **UN/EDIFACT, ISO/IEC 14888, ISO/IEC 9796-4, ISO/IEC 14946.** (International Standards Organization).
- **ATM.** Forum (Asynchronous Transport Mode).
- **WAP.** (Wireless Application Protocol).

En el comercio electrónico:

- **FSTC.** (Financial Services Technoly Consortion).
- **OTP 0.9** (Open Trading Protocol).
- **SET.** (Secure Electronics Transactions). En Internet **IETF** (The Internet Engineering Task Fource) .
- **IPSec** (Internet Protocol Security Protocol).

Los **CCE** están reemplazando a las aplicaciones que tienen implementado **RSA**, estas definen también esquemas de Firma Digital, intercambio de claves simétricas y otros.

3.2.5. VENTAJAS DE LOS SISTEMAS CRIPTOGRÁFICOS DE CLAVE PÚBLICA FRENTE A LOS DE CLAVE PRIVADA.

La principal ventaja de los Sistemas de Clave Pública o Asimétricos frente a los Sistemas de Clave Privada o Simétricos es que la clave pública y el algorítmo de cifrado son o pueden ser de dominio público y que no es necesario poner en peligro la clave privada en tránsito por los medios inseguros, ya que está siempre oculta y en poder únicamente de su propietario.

Como desventaja, los Sistemas de Clave Pública dificultan la implementación del sistema y son mucho más lentos que los Sistemas de Clave Privada.

Generalmente, y debido a la lentitud de proceso de los Sistemas de Llave Pública, estos se utilizan para el envío seguro de Claves Simétricas, mientras que los Sistemas de Llave Privada se usan para el envío general de los datos encriptados.

3.2.6. RESUMEN.

Para sintetizar la explicación y entender el funcionamiento básico de un Criptosistema Asimétrico podemos representarlo gráficamente con los siguientes pasos:

CLAVE PÚBLICA:

Tanto el **Emisor** como el **Receptor** posee dos claves complementarias, la clave pública y la clave privada, lo que se codifica con una de ellas, solo puede ser decodificado con la otra clave.
El **Emisor** y el **Receptor**, deben custodiar su clave privada y su clave pública, la clave privada sólo debe ser conocida por él (Receptor), y la clave pública puede ser conocida por cualquier persona.
Los pasos a realizar para llevar a cabo un envío de mensajes son:

Emisor:

1. El **Emisor** dispone de clave pública y de clave privada.

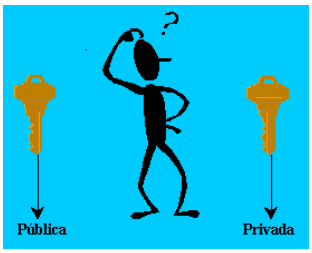


Figura 3.25

2. El **Emisor** debe conocer la clave pública del **Receptor**.

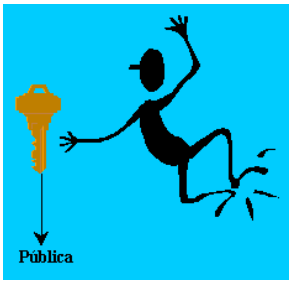


Figura 3.26

3. El **Emisor** escribe el mensaje a enviar y lo cifra mediante la clave pública del **Receptor**.

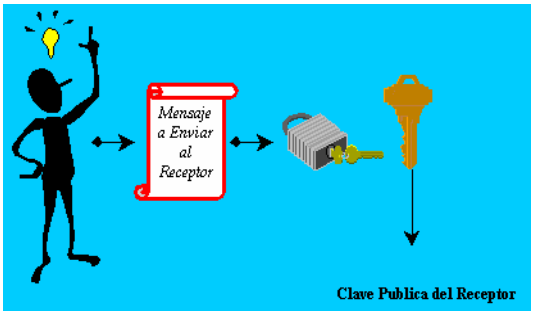


Figura 3.27

4. El **Emisor** envía el mensaje al **Receptor**, también le puede enviar su propia clave pública

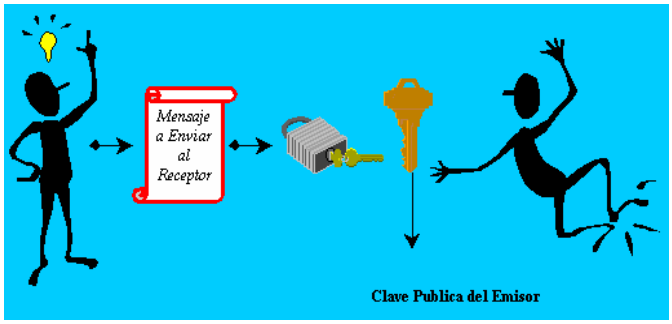


Figura 3.28

Receptor :

1. Recibe el mensaje del **Emisor** codificado.

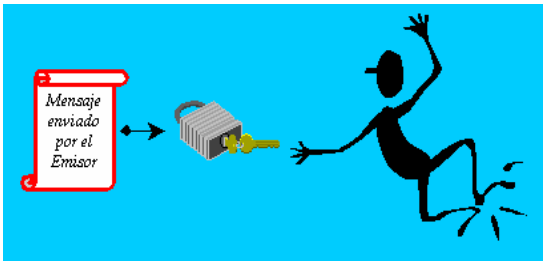


Figura 3.29

2. Descifra el mensaje recibido con su clave privada.

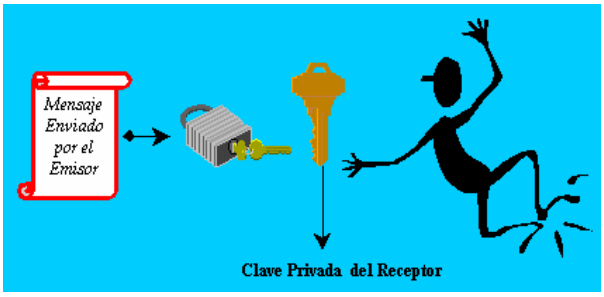


Figura 3.30

3. Obtiene el mensaje original.

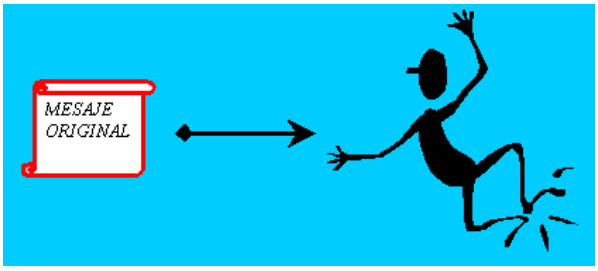


Figura 3.31

Conclusiones:

1. El Emisor cifra el mensaje con la clave pública del Receptor.
2. El Receptor descifra el mensaje con su clave primaria.
3. Distribución pública de la clave pública.
Tanto el emisor como el receptor sólo disponen de una clave privada y una pública.

Capítulo

4

APLICACIONES DE SEGURIDAD

Existe gran variedad de **Aplicaciones de Seguridad** que dan soporte a la ***Criptografía Moderna***, y que en materia de seguridad vienen a fijar las reglas del juego en el mercado de las Transacciones Electrónicas, a continuación se citarán las más populares y de mayor uso en los continentes de América y Europa.

4.1. INTRODUCCIÓN.

Un **Protocolo de Seguridad,** es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo específico y que usan esquemas de Seguridad Criptográfica.
Además de lo anterior dicho podría decirse que un **Protocolo de Seguridad** es un conjunto de normas que regulan el intercambio dinerario entre las entidades participantes en una Transacción de Comercio Electrónico, garantizando el buen fin de la operación.

4.2. SSL (*SECURE SOCKETS LAYER*).

4.2.1. INTRODUCCIÓN.

La incorporación de Técnicas Criptográficas a los sistemas de comunicación ha propiciado que la ***Criptografía*** y sus métodos dejen de ser teorías oscuras, conocidas por unos pocos, para convertirse en elementos tecnológicos de dominio público (cada vez es más común el establecimiento de servidores web seguros basados en **SSL**). Sin embargo, dicha difusión ha provocado también que muchos de los términos relacionados con el tema sean mal interpretados y confundidos, tanto por los usuarios como por los profesionales de la informática. La confusión suele venir propiciada, normalmente, por el conocimiento parcial de la compleja teoría que envuelve a la ***Criptografía.***

El protocolo **SSL** es un sistema diseñado y propuesto por ***Netscape Communications Corporation***. Es un protocolo usado por los Estados Unidos que carece de Certificados Digitales e incluye otras particularidades derivadas de la especial legislación estadounidense, sobre todo en materia de identidad intimidad de las personas.

Se encuentra en la pila OSI y ofrece seguridad entre los niveles de TCP/IP y de los protocolos HTPP, FTP, SMTP, además ofrece los servicios de Autenticación del servidor y del cliente y de Confidencialidad. El mensaje se cifra con una clave simétrica seleccionada aleatoriamente.
Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algorítmo de cifrado simétrico, típicamente el **RC4** o **IDEA**, y cifrando la clave de sesión de **RC4** o **IDEA** mediante un algorítmo de cifrado de clave pública, típicamente el **RSA**. La clave de sesión es la que se utiliza para cifrar los datos que vienen de él y van al servidor seguro.
Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. **MD5** se usa como algoritmo de Hash.

Es el protocolo de comunicación segura más conocido y usado actualmente, **SSL** actúa en la capa de comunicación y es como un túnel que protege a toda la información enviada y recibida.
Con **SSL** se puede usar diferentes algorítmos para la diferentes aplicaciones, por ejemplo **DES, TDES, RC2, MD5, SHA-1, DH y RSA.** Cuando una comunicación está bajo **SSL,** la información que es cifrada es:

- El URL del documento requerido.
- El contenido del documento requerido.
- El contenido de cualquier forma requerida.
- Los "cookies" enviados del browser al Server.
- Los "cookies" enviados del Server al browser.
- El contenido de las cabeceras de los http.

En concreto, nos vamos a centrar en un error cometido frecuentemente, como es el tamaño y tipo de las claves, aplicado en este caso al protocolo **SSL**.

SSL es un protocolo de comunicación que proporciona principalmente tres servicios básicos de Seguridad: Confidencialidad, Autenticación e Integridad. Con el fin de garantizar dichos servicios, **SSL** hace uso tanto de la Criptografía Asimétrica (basada en la existencia de un par de claves, la pública y la privada) como de la Criptografía Simétrica (basada en la utilización de una única clave secreta).
La justificación de dicha combinación viene dada por cuestiones de eficiencia, puesto que las transformaciones criptográficas (operaciones de cifrado y descifrado) realizadas mediante técnicas de Criptografía Asimétrica son del orden de diez mil veces más lentas que las realizadas con Criptografía Simétrica.

SSL negocia en una primera fase utilizando Criptografía Asimétrica (RSA), y cifra posteriormente la comunicación utilizando Criptografía Simétrica (RC4, RC5, IDEA...).

La confusión viene provocada por esta combinación de técnicas que utiliza distintos tipos de claves. Las claves empleadas en Criptografía Asimétrica tienen justificación matemática, mientras que las que se utilizan en Criptografía Simétrica suelen ser simples cadenas de bytes aleatorios. Esta diferencia de contenido hace que no sea comparable el tamaño de las claves simétricas y asimétricas.

El tamaño de Clave Simétrica suele oscilar entre los 40 y los 128 bits. Las claves de 40 bits (como las utilizadas por Netscape en su versión de exportación), pueden romperse en cuestión de horas, mientras que las claves de 128 bits son irrompibles actualmente.
Otros tamaños estándar de clave son 56 bits (DES), que tampoco proporciona seguridad hoy en día, ya que puede romperse en cuestión de días, u 80 bits (SKIPJACK). La seguridad completa sólo se consigue actualmente utilizando claves no inferiores a 80 bits.

Por otro lado, *el tamaño de Clave Asimétrica* oscila entre los 512 bits y los 4096 bits. Las claves de 512 bits (también utilizadas por Netscape en su versión de exportación), han dejado de considerarse seguras últimamente, y no se recomienda el uso de claves inferiores a 768 bits.

La confusión es lógica, pero puede evitarse fácilmente identificando los principales algorítmos criptográficos y ubicándolos dentro de su categoría (simétricos o asimétricos).

4.2.2. TIPOS DE SSL.

Como vimos anteriormente **SSL**, proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP.
Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo **SSL Record**, situado encima de TCP. Será el software de alto nivel, Protocolo **SSL Handshake,** quien utilice el Protocolo **SSL Record** y el puerto abierto para comunicarse de forma segura con el cliente.

a. El Protocolo SSL Handshake:

Durante el protocolo **SSL Handshake**, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases (de manera muy resumida):

- La fase **Hola**, usada para ponerse de acuerdo sobre el conjunto de algorítmos para mantener la intimidad y para la autenticación.

- La fase de **Intercambio de Claves**, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.
- La fase de **Producción de Clave de Sesión**, que será la usada para cifrar los datos intercambiados.
- La fase de **Verificación del Servidor**, presente sólo cuando se usa **RSA** como algorítmo de intercambio de claves, y sirve para que el cliente autentique al servidor.
- La fase de **Autenticación del Cliente**, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).
- La fase de **Fin**, que indica que ya se puede comenzar la sesión segura.
-

b. El Protocolo SSL Record:

El protocolo **SSL Record** especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

- MAC - DATA, el código de autenticación del mensaje.
- ACTUAL - DATA, los datos de aplicación a transmitir.
- PADDING - DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

4.2.3. COMUNICACIÓN SEGURA CON SSL.

El procedimiento que se lleva a cabo para establecer una comunicación segura con **SSL** es el siguiente, veámoslo con un ejemplo:

1. El cliente (browser) envía un mensaje de saludo al Server "ClientHello.
2. El Server responde con un mensaje "ServerHello".
3. El Server envía el certificado.
4. El Server solicita el certificado del cliente.
5. El cliente envía su certificado: si es valido continua la comunicación si no para, o sigue la comunicación sin certificado del cliente.
6. El cliente envía un mensaje "ClientKeyExchage" solicitando un intercambio de claves simétricas si es el caso.
7. El cliente envía un mensaje "CertificanteVerify" si se ha verificado el certificado del Server, en caso de que el cliente este en estado de autenticación.
8. Ambos clientes y Server envían un mensaje "ChangeCipherSpec" que significa el comienzo de la comunicación segura.
9. Al terminar la comunicación ambos envían el mensaje "finished" con lo que termina la comunicación segura, este mensaje consiste en un intercambio de funciones hash de toda la conversación, de manera que ambos están seguros que los mensajes fueron recibidos intactos.

La versión más actual del **SSL** es la v3, existen otro protocolo parecidos al **SSL** solo que es desarrollado por IETF se denomina **TLS** (Transport Layer Segurity Protocol), y difiere en que usa un conjunto un poco más amplio de algorítmos

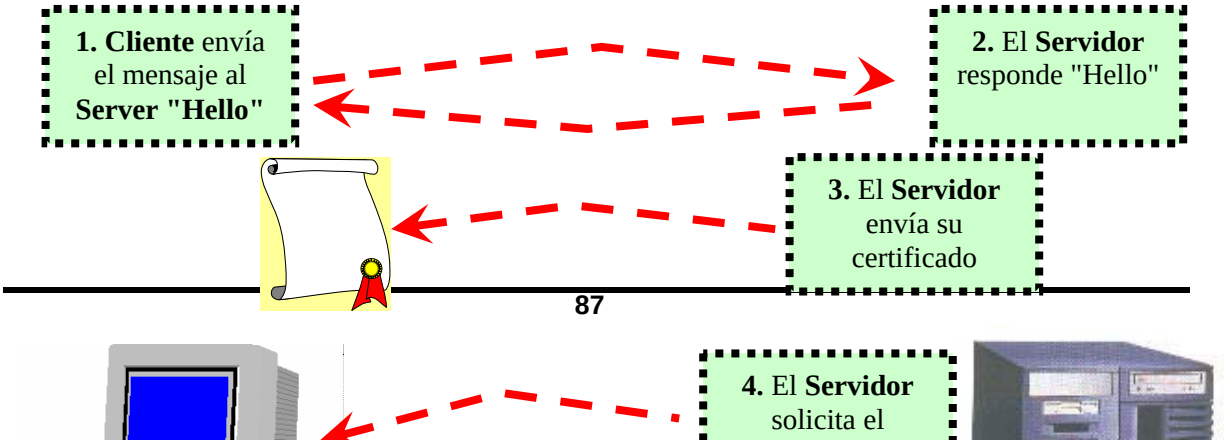
criptográficos. Por otra parte existe también **SSL Plus**, un protocolo que extiende la capacidad del **SSL** y tiene por mayor característica que es interoperable con **RSA, DSA/DH y CCE** (Criptografía con Curvas Elípticas).

NOTA:

El protocolo **SSL** lo vemos integrado en el browser del Netscape y hace su aparición cuando el candado de la barra de herramienta se cierra y también si la dirección de Internet cambia de **http a https.**

Reflejando gráficamente los anteriores pasos los describiríamos así.

FIGURA 4.1



4.3. PGP (*PRETTY GOOD PRIVACY*).

4.3.1. INTRODUCCIÓN.

En 1991, se extendió por EE.UU. el rumor de que el Gobierno pretendía prohibir el uso de ***Criptografía*** en líneas de comunicación. A raíz de aquel rumor, Phil Zimmermann crea el programa **PGP**. Desde entonces, multitud de programadores

de todo el mundo, en la mejor tradición de Internet, han contribuido a la depuración, portabilidad a diferentes sistemas operativos y mejora del paquete.

Todo el paquete y su documentación se hicieron públicos y libremente distribuibles. En junio de 1991, alguien envía a varios grupos de noticias de USENET una copia del programa, con lo que empezó a difundirse y utilizarse fuera de EE.UU.
Llegado a este punto, Zimmermann se enfrentó a dos problemas. Por una parte, la distribución de este programa fuera de EE.UU. transgredía, supuestamente, las normativas de exportación de ***Criptografía***. Por otra, **RSA Data Security, Inc.** le demandó por el uso inautorizado del algoritmo **RSA**. En 1996, Phill Zimmermann fue absuelto de ambas demandas, creando a continuación **PGP Inc.,** para distribuir nuevas versiones del programa. En la actualidad, **PGP Inc**., ha sido absorbida por *Network Associates Inc*. (que posee también, entre otras, a *McAffee* o a TIS, la creadora de *Gauntlet*).
En julio de 1997 apareció la versión internacional de PGP 5.0, la cual añade nuevos algorítmos a los ya citados:

- El algoritmo **ElGamal** como algorítmo de Clave Asimétrica (alternativa al algoritmo **RSA**).
- **Triple-DES y CAST** como Algoritmos de Clave Simétrica.
- **SHA-1** como función resumen.

PGP ha sido, y es, un hito dentro de Internet. Se trata de un sistema completo que proporciona Confidencialidad (permite a un usuario enviar un mensaje cifrándolo, con la garantía de que solamente su destinatario podrá leerlo), Autenticación (utilizando firmas digitales, un mensaje sólo puede ser leído por su destinatario si su remitente es quien dice ser), Integridad (la firma antes citada depende no sólo del remitente sino también del contenido del mensaje. Por tanto, si el mensaje es alterado, la firma no es válida) y sirve básicamente para el intercambio Correo Electrónico (E-mail).

4.3.2. IMPLEMENTACIÓN.

En orígen, el mensaje o fichero que queremos proteger va pasando por diferentes bloques que lo van transformando:

- ***Compresión***: utilizando el algorítmo *pkzip*.
- ***Firma Digital***: Se crea una Firma Digital del mensaje, utilizando la función de *hash* MD5. Dicha firma es cifrada con la clave privada del remitente usando el algoritmo **RSA** y añadida al mensaje.
  El usuario puede elegir la longitud de la clave asimétrica, y por tanto su grado de seguridad. Ahora bien, cuanto más larga sea la clave, más lento será el proceso.
- ***Cifrado:*** El cifrado se hace usando **IDEA** (criptografía de clave simétrica). La clave simétrica es generada aleatoriamente y transmitida al destinatario cifrada con su clave pública (algorítmo **RSA** de nuevo).
- ***Compatibilización:*** con el protocolo de transmisión de correo electrónico.

- ***Segmentación:*** En la recepción se realiza el proceso inverso, con lo que se obtiene el mensaje original, asegurando los requerimientos buscados.

La característica más peculiar de **PGP** es su Estructura de Certificación. En lugar de adoptar una estructura jerárquica con un notario de confianza en la raíz del árbol, **PGP** soporta una estructura más o menos arbitraria. Los usuarios reciben claves públicas de otros usuarios y las añaden a su propio anillo de claves (*keyring*), indicando el grado de confianza que tienen en dichas claves (lo seguro que están de que dichas claves pertenezcan a quien dice ser su propietario). Aunque muy similar a los mecanismos sociales de confianza entre las personas, es aquí, probablemente, donde se encuentra el talón de Aquiles de **PGP**.

PGP es tan popular, lo cual conlleva a que las claves de éxito sean las siguientes:
1. El paquete **PGP** está en el dominio público.
2. El paquete incluye código fuente y documentación. Cualquiera puede verificar sus fundamentos e incorporar mejoras.
3. No está bajo el control de ningún gobierno o empresa.
4. Se encuentra disponible para multitud de plataformas.
5. Está basado en algorítmos seguros y ampliamente probados.

Para usar **PGP,** hay que comenzar generando un par de claves, una pública y otra privada, siendo posible en ese momento la elección de la longitud de clave deseada. También hay que fijar una clave personal, que se usará luego para proteger la llave privada de miradas indiscretas. Las claves pública y privada las genera automáticamente el algorítmo, mientras que la personal de protección la elige el usuario.
Una vez generadas las claves, la privada se encripta con la personal mediante un algoritmo simétrico, siendo posteriormente necesario desencriptarla cada vez que deseemos usarla.

En cuanto a la clave pública, se deposita en un fichero especial, de tipo ASCII (sólo texto), denominado **Certificado de Clave**, que incluye el identificador de usuario del propietario (el nombre de esa persona y algún dato único, como su dirección de e-mail), un sello de hora del momento en el que se generó el par de llaves y el material propio de la clave.
Cuando se desea mandar un correo o fichero encriptado, **PGP** lo encripta usando un sistema simétrico, generalmente **IDEA o DES**, usando una clave aleatoria, que posteriormente es encriptado con **RSA**. Se envían el documento cifrado con la clave aleatoria y está encriptada con la llave RSA privada del destinatario.
Cuando éste recibe el correo y desea desencriptarlo, su programa **PGP** primero descifra la clave simétrica con su llave privada **RSA**, y luego descifra el documento usando la clave desencriptada.

Este proceso lo ilustramos en el siguiente gráfico.

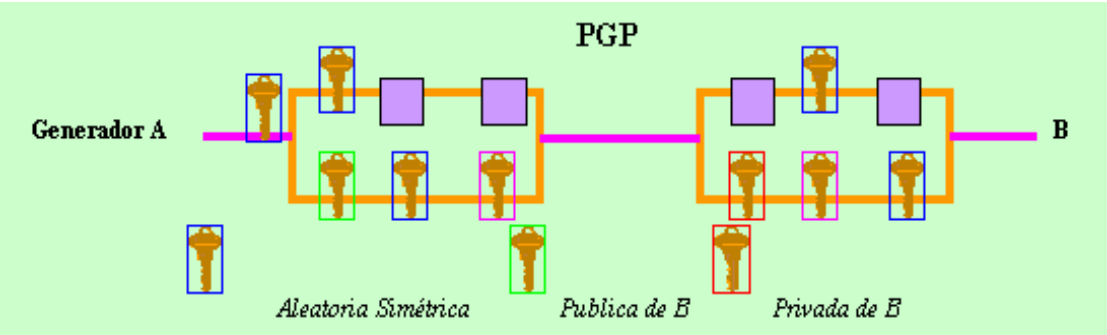


Figura 4.2

Normalmente el sistema **PGP** viene implementado mediante alguna aplicación específica, que se instala en el computador del usuario. Esta aplicación se integra perfectamente con los programas de correo más comunes, permitiendo al usuario el uso directo del sistema **PGP**, con tan sólo pulsar los botones que aparecerán en la barra de menús de la aplicación de correo.

Para desencriptar un mensaje basta con seleccionar el icono correspondiente (o mediante el botón derecho del ratón), siendo necesario en ese momento introducir la clave personal, para que el programa pueda acceder a la clave pública encriptada. Para firmar un correo se procede de forma análoga.
En el caso de querer enviar un e-mail encriptado a otra persona, es necesario en primer lugar que la misma tenga un programa **PGP** instalado, y después que tenga la llave pública del destinatario.
Si es así, basta con seleccionar la opción correspondiente en el menú, con lo que se nos pedirá la clave pública del destinatario, y el programa se encargará de todo lo demás.

Si tenemos la necesidad de cifrar mensajes para muchos destinatarios diferentes nos encontraremos con el problema de gestionar los distintos ficheros de llave pública. Para ello, los programas **PGP** facilitan el denominado **llavero**, que es un pequeño módulo de software que se encarga de administrar dichos ficheros. Cuando recibamos uno de ellos, basta con colocarlo en el llavero para tenerlo disponible siempre que deseemos.
Puede interesarnos en un momento dado enviar un correo cifrado o vernos en la necesidad de comprobar la autenticidad de un correo que nos llega firmado por parte de un destinatario del que no conocemos su llave pública.

Podemos obtener ésta dirigiéndonos directamente a dicha persona y pidiéndosela, pero para facilitar esta tarea, y puesto que el objeto de las llaves públicas es ser difundidas lo más posible, se han habilitado diferentes servidores que poseen bases de datos con las claves públicas de los usuarios de **PGP**.
Basta acudir a los mismos y solicitar el fichero de clave correspondiente a la persona que nos interesa.

El uso de **RSA** ha hecho que la difusión y uso de **PGP** se haya visto sujeta a las controversias provocadas por la necesidad de una licencia de exportación, lo que le ha costado a Phill Zimmermann ser procesado por ello.

4.3. SET (*SECURE ELECTRONIC TRANSACTIONS*).

4.3.1. INTRODUCCIÓN.

Durante 1995, las grandes compañías mundiales de tarjetas de crédito presentaron sendas proposiciones de comercio electrónico, con vistas a incorporar los prácticamente universales medios de pago electrónicos al mundo de Internet. *Visa*, en colaboración con *Microsoft*, desarrolló una especificación completa, la *Secure Transactions Technology* (STT). Por otra parte, *MasterCard*, en asociación con *IBM*, *Netscape* y *CyberCash* patrocinó una especificación conocida como *Secure Electronic Payment Protocol* (SEPP). Ambas especificaciones se basaban en el uso de **Criptografía de Clave Pública y Certificados.**

Ante la previsible guerra que se adivinaba, ambos gigantes, junto con los consorcios que les apoyaban, decidieron asociarse y presentaron, en febrero de 1996, una especificación abierta para conseguir la protección de los pagos hechos mediante tarjeta de crédito en cualquier red insegura y, específicamente, en Internet. *American Express* se unió al consorcio poco después de la publicación del primer borrador. Se consideró que las primeras implementaciones se desarrollarían durante el año 1997. A finales de 1998, la mayoría de los bancos adoptan el protocolo y lo incluyen en sus pasarelas de pago. Al igual que las transacciones tradicionales realizadas con tarjeta de crédito van acompañadas de un número de autenticación de la operación, éste sistema es una Certificación Digital de la transacción electrónica, un sello de confianza que garantiza la viabilidad de la operación entre consumidor y comerciante.

El sistema **SET** garantiza la verificación del titular, la integridad, el no repudio y, sobre todo, la privacidad de la operación, cada interviniente tiene acceso únicamente a la información que tiene estipulada.

La previsión futura es permitir el pago con tarjetas tipo SmartCard o TID (tarjeta identificador digital), que incluyen la información actual de la tarjeta de pago y el propio certificado **SET** del usuario; para ello hay que dotar al computador personal del correspondiente lector de tarjetas.

4.3.2. OBJETIVOS DE SET.

Los objetivos que se perseguía con el desarrollo del protocolo **SET** fueron los siguientes:

1. Definir un estándar único para efectuar transacciones a través de Internet, evitando la competencia entre diferentes estándares auspiciados por distintas empresas y consorcios.
2. Este estándar debe ser similar y compatible con los sistemas de pago mediante tarjeta existentes en la actualidad.
3. Proveer la Autenticación de todas las partes implicadas en una transacción.
4. Mantener la confidencialidad de la información intercambiada, de forma que cada parte no tenga acceso a más información que la estrictamente necesaria para llevar a cabo su función en la transacción.
5. Mantener la Integridad de la información implicada en los pagos.
6. La búsqueda de independencia de plataformas y navegadores.

4.3.3. FUNCIONAMIENTO DE SET.

El proceso de **SET** es más o menos el siguiente:

1) **El cliente inicializa la compra:** Consiste en que el cliente usa el browser para seleccionar los productos a comprar y llena la forma de orden correspondiente. **SET** comienza cuando el cliente hace clic en "pagar" y se envía un mensaje de iniciar **SET**.

2) **El cliente usando SET envía la orden y la información de pago al comerciante:** El software **SET** del cliente crea dos mensajes uno conteniendo la información de la orden de compra, el total de las compras y el número de orden. El segundo mensaje contiene la información de pago, es decir, el numero de la tarjeta de crédito del cliente y la información del banco emisor de la tarjeta. El primer mensaje es cifrado usando un sistema simétrico y es empaquetado en un sobre digital que se cifra usando la clave publica del comerciante. El segundo mensaje es cifrado pero usando la clave publica del banco (esto previene que el comerciante tenga acceso a los números de tarjetas de los clientes), finalmente el cliente firma ambos mensajes.

3) **El comerciante pasa la información de pago al banco:** El software **SET** del comerciante genera un requerimiento de autorización, este es comprimido (con una función hash) y firmado por el comerciante para probar su identidad al banco del comerciante, además de ser cifrado con un sistema simétrico y guardado en un sobre digital que es cifrado con la clave pública del banco.

4) **El banco verifica la validez del requerimiento:** El banco descifra el sobre digital y verifica la identidad del comerciante, en el caso de aceptarla descifra la información de pago del cliente y verifica su identidad. En tal caso genera un requerimiento de autorización lo firma y envía al banco la tarjeta que genera la tarjeta del cliente.

5) **El emisor de la tarjeta autoriza la transacción:** El banco del cliente (emisor de la tarjeta) confirma la identidad del cliente, descifra la información recibida y verifica la cuenta del cliente en caso de que no haya problemas aprueba el requerimiento de autorización, lo firma y lo regresa al banco del comerciante.
6) **El banco del comerciante autoriza la transacción:** Una vez recibida la autorización del banco emisor, el banco del comerciante autoriza la transacción la firma y la envía al servidor del comerciante.

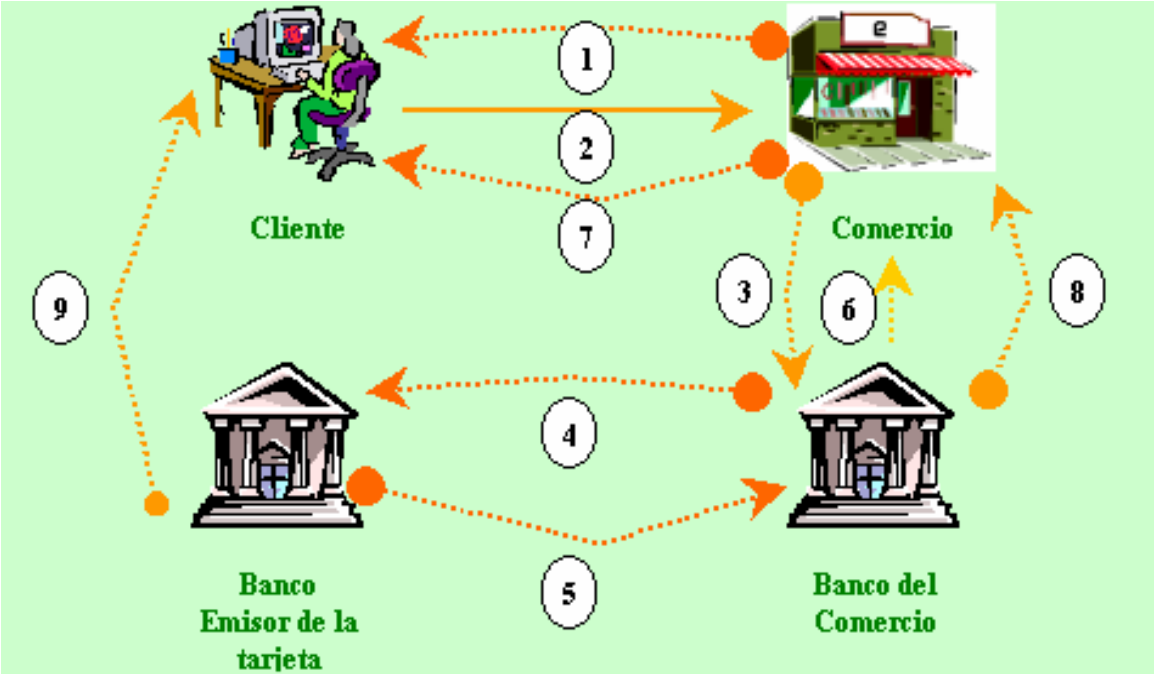
7) **El servidor del comerciante complementa la transacción:** El servidor del comerciante dá a conocer de que la tarjeta fue aprobada, muestra al cliente la conformidad de pago y procesa la orden que pide el cliente terminando la compra cuando se les son enviados los bienes que compró el cliente.

8) **El comerciante captura la transacción:** En la fase final de **SET** el comerciante envía un mensaje de "captura" a su banco, esto confirma la compra y genera el cargo a la cuenta del cliente, así como acreditar el monto a la cuenta del comerciante.

9) **El generador de la tarjeta envía el aviso de crédito al cliente:** El cargo de **SET** aparece en estado de cuenta del cliente que se le envía mensualmente.

SET requiere un Certificado Digital en cada paso de autenticación y usa dos pares de claves, una para el cifrado del sobre digital y otra para la firma, (SSL solo usa un par de claves), actualmente **SET** usa la función hash AHA-1,DES y **RSA** de 1024 bits, estos parámetros fueron tomados para ser compatible con los certificados existentes, aunque el piloto de **SET** usa el sistema asimétrico de cifrado con Curvas Elípticas y se piensa que soporta curvas elípticas en la próxima versión de **SET**.
Lo anteriormente expuesto se representaría gráficamente así:

Figura 4.3



4.3.4. CONCLUSIONES.

El protocolo **SET** es la respuesta de las grandes compañías mundiales de medios de pago al desafío del **Comercio Electrónico**. Cuentan como aliados a los grandes del mundo de la informática. Como no podía ser menos, resuelve de forma completa los problemas de autenticación de las partes, confidencialidad e integridad de los datos y no repudio.

Hay que considerar que **SET**, por la propia naturaleza de los pagos (mediante tarjeta de crédito) no parece muy adecuado para su utilización en el microcomercio, un segmento nada desdeñable de las potenciales transacciones que podrán llevarse a cabo.

Aún no se puede afirmar el aceptar a este protocolo como un estándar en el futuro ya que no se cuentan con los datos suficientes o si, por el contrario, no conseguirá una implantación mayoritaria, consecuencia de la tardanza en ponerse en funcionamiento y de su "pesadez". Una alternativa podría ser la adopción de una versión aligerada del protocolo. En todo caso, cualquier desarrollo en el campo del

Comercio Electrónico en la actualidad no puede perder de vista el "inminente" aterrizaje de este estándar.

Este protocolo está especialmente diseñado para asegurar las transacciones por Internet que se pagan con tarjetas de crédito. Esto es debido a que una gran cantidad de transacciones de compras por Internet son efectuadas con tarjeta de crédito, que por otro lado, **SSL** deja descubierto alguna información sensible cuando se usa para lo mismo.

La principal característica de **SET**, es que cubre estos huecos en la seguridad que deja **SSL**.

Por ejemplo: con **SSL** solo protege el número de tarjeta cuando se envía del cliente al comerciante, sin embargo no hace nada para la validación del número de tarjeta, para chequear si el cliente esta autorizado a usar ese número de tarjeta, para ver la autorización de la transacción del banco del comerciante. Además que el comerciante puede facilitar guardar el número de tarjeta del cliente. En fin todas estas debilidades son cubiertas por **SET**, esto permite dar seguridad tanto al cliente, al comerciante como al banco emisor de la tarjeta y al banco del comerciante.

Habiendo abordado a fondo estos tres protocolos de gran importancia en el uso de ***Criptografía Moderna***, podemos citar algunas diferencias entre ellos, principalmente entre **SSL (*Norteamericano*)** y **SET (*Europeo*)** como lo veremos a continuación:

- **SSL** tiene mayor facilidad de uso, está soportado de forma estándar por la mayoría de navegadores, mientras que la utilización de **SET** es más compleja, necesita una gestión de certificados que lo hacen menos versátil.

- En cuanto a la seguridad, **SET** es mucho más seguro, la garantía de la integridad y confidencialidad de la información es prácticamente absoluta, la realización de una transacción **SSL** se puede considerar equivalente a realizar una compra por correo o por teléfono, enviando los datos de la tarjeta de crédito con pocas garantías de seguridad, lo que conlleva mayor riesgo de rechazo de operaciones por los clientes. En cambio, una transacción **SET** equivale a una compra personal, entregando la tarjeta de crédito en mano, lo que representa una aceptación total por el cliente.

- El coste del sistema **SSL** es más reducido que **SET**, puesto que está integrado en el navegador y carece de certificados.

- Al estar incluido en los navegadores, el ámbito de **SSL** es universal, mientras que **SET** está aún en fase de implantación, aunque en Europa es prácticamente el único que se utiliza y se va introduciendo poco a poco en el mercado norteamericano.

Estos y cualquier **Protocolo de Seguridad** procuran resolver algunos de los problemas de seguridad como la Integridad, Confidencialidad, Autenticación y el No rechazo, mediante sus diferentes características.

Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, como lo es: robar, cambiar, leer información no autorizada y todo lo que se considere no autorizado por los usuarios de una comunicación por la red.

Capítulo 5

CERTIFICADOS DIGITALES

5.1. INTRODUCCIÓN.

Uno de los problemas que surgen en Internet es el de la identificación de las personas o entidades.
Por ejemplo: ¿Cómo asegurarnos que una clave pública que hemos encontrado en Internet pertenece realmente a quién dice pertenecer?
Una posible solución es la utilización de un ***Certificado Digital*** que es un fichero digital intransferible y no modificable, emitido por una tercera parte de confianza que es la **Autoridad Certificadora (AC)**, que asocia a una persona o entidad una clave pública.

El nacimiento del ***Certificado Digital*** fue a raíz de resolver el problema de administrar las claves públicas y que la identidad del dueño pudiera ser falsa. La idea es que una tercera entidad intervenga en la administración de las claves públicas y asegure que las claves públicas tengan asociadas un usuario claramente identificado. Esto fue inicialmente planteado por Kohnfelder del MIT en su tésis de Licenciatura.

Dando una definición más exacta de lo que son los **Certificados Digitales** podríamos decir que son: *Documentos digitales que atestiguan que una clave pública corresponde a un individuo o entidad determinada.* De este modo evitamos que intrusos utilicen una combinación de claves asegurando ser otra persona.
En su forma más simple, un **certificado** consiste en una clave pública y el nombre de su propietario. Este certificado es firmado por una autoridad de certificación (Certification Authority, CA), cuya clave pública es fácilmente verificable.
.
Los **Certificados Digitales** tienen una similitud con las licencias de conducir, las primeras permiten viajar por las carreteras, los **Certificados Digitales** permiten navegar por la red Internet, la principal característica es que, da identidad al usuario y puede navegar con seguridad.

De igual forma que la licencia para conducir o un pasaporte sirve para dar identidad a quien la porta en ciertos casos, el ***Certificado Digital*** dá identidad a una clave pública.

5.2. PARTES DE UN CERTIFICADO DIGITAL.

Las tres partes de un ***Certificado Digital*** son:

1. Una clave pública.
2. La identidad del implicado: nombre y datos generales.
3. La firma privada de una tercera entidad llamada **Autoridad Certificadora** que todos reconocen como tal y que valida la asociación de la clave pública en cuestión con el tipo que dice ser.

En la actualidad casi todas las aplicaciones de Comercio Electrónico y transacciones seguras requieren un ***Certificado Digital***, se ha propagado tanto su uso que se tiene un formato estándar de ***Certificado Digital***, este es conocido como x509 v.3.

Un ***Certificado Digital*** que siga el estándar X509v3, utilizado por los navegadores, contiene la siguiente información:

- **Identificación del titular del certificado:** Nombre, dirección, etc.
- **Clave pública del titular del certificado.**
- **Fecha de validez.**
- **Número de serie.**
- **Identificación del emisor del certificado.**

Algunos de los datos más importantes de este formato son.

Gráficamente un ***Certificado Digital*** se vería así:

Versión: 1,2 o3
Numero de Serie:
Emisor del Certificado: VeriMex.
Identificador del Algoritmo usado en la firma: RSA, DSA o CE.
Periodo de Validez: De Enero 2002 a Diciembre 2002
Sujeto: James Alan Hetfield.
Información de la clave publica del sujeto: La clave, longitud y demás parámetros
Algunos datos opcionales, extensiones que permite la v3:
Firma de la Autoridad Certificadora:

Figura 5.1

Un ***Certificado Digital*** entonces se reduce a un archivo de uno o dos **k** de tamaño, que autentica a un usuario de la red.

Para verificar que el certificado es correcto deberíamos hacernos con el ***Certificado Digital*** emitido para dicha **AC** por una segunda **AC**. Para verificar la veracidad de este segundo certificado deberíamos obtener el ***Certificado Digital*** emitido para segunda AC por una tercera **AC**. Como este proceso podría eternizarse, existen las llamadas autoridades raíz que firman sus propios certificados, un ejemplo de autoridad raíz es **VeriSing.**

Aparte de los datos del emisor y del propietario del certificado éste puede contener información referente a las limitaciones que se hayan establecido para su uso: e-mail, WWW, etc.
Como podemos ver en el certificado del ejemplo, no existe ninguna referencia a algorítmos de clave secreta (Ver Figura 5.1).

5.3. SERVICIOS QUE OFRECE UNA AUTORIDAD CERTIFICADORA (AC).

Una **AC**, además de emitir certificados, debe ofrecer los servicios siguientes:

- ***Búsqueda de Certificados:*** Una persona puede querer buscar el certificado referente a otra persona o entidad.
- ***Revocación:*** Si un certificado se pierde, el titular debe poder informar a la **AC** para que lo anule y emita otro. También si la clave privada ha quedado comprometida debe ser posible su revocación.
- ***Suspensión:*** La **AC** debe suspender la validez de un certificado si se hace un uso anormal de él.
- ***Estado del Certificado:*** Las personas a las que se les presenta un certificado deben de poder comprobar que no ha sido revocado o suspendido.

Actualmente aún se está desarrollando el marco legal que regule el reconocimiento de la validez legal de las Firmas Digitales y cuáles han de ser los requerimientos mínimos que han de reunir las Autoridades Certificadoras para ser reconocidas como tales.

No podemos olvidar que la **Autoridad de Certificación** es la responsable, en última instancia, de todo el proceso, con una serie de responsabilidades legales y que basa su "negocio" en la credibilidad que inspire en sus potenciales clientes. Una **Autoridad de Certificación** con autentificaciones erróneas no tendrá más remedio que cerrar ya que los usuarios no considerarán sus certificados de la suficiente "calidad".

5.4. CICLO DE VIDA DE UNA CLAVE.

Las claves deben tener una fecha de expiración. De esta forma, es más difícil que los algorítmos que las utilizan sufran algún ataque. Para dar un ejemplo:

1. Un intruso puede almacenar texto cifrado con objeto de averiguar la clave. Si la clave expira, el texto cifrado almacenado ya no sirve.
2. La clave puede haber sido ya averiguada, pero el ataque puede ser, hasta la fecha, pasivo.
3. Las técnicas de análisis de los algorítmos criptográficos (como resolver el problema de la factorización en el caso del algoritmo RSA, por ejemplo) avanzan constantemente. El tamaño de las claves debe ir incrementándose para evitar este tipo de riesgos (por tanto hay que cambiar la clave).

Cuando una clave ha sido averiguada por intrusos, se dice que ha sido comprometida.

El ciclo de vida de una clave incluye los siguientes períodos:

a. Generación y, quizás, Registro de la clave o par de claves.
La clave o par de claves debe ser generada por su propietario o por la entidad que vaya a utilizar la/s clave/s para proteger sus comunicaciones con el usuario. Un problema frecuente radica en que los algorítmos generadores aleatorios de claves no son suficientemente "buenos". Si la clave es utilizada con algorítmos de criptografía de clave asimétrica, la clave pública puede ser registrada (generando un certificado).

b. Distribución de las claves.
En el caso de criptografía de clave simétrica, la clave debe ser entregada al interlocutor de forma que no pueda ser interceptada por terceros. En caso de utilizar claves asimétricas, la distribución de esta clave está libre de problemas. Sin embargo, debe poder asegurarse que la clave corresponda a quien dice ser su propietario (mediante un certificado, o bien obteniendo la clave de una organización en la que se tenga plena confianza).

c. Emisión y **expiración**.
La fecha de emisión determina a partir de qué instante va a ser válida la clave. En general, se trata del momento en el que ha sido generada (o certificada, en su caso). La expiración puede tener lugar al final de una comunicación concreta o en una fecha determinada. En el caso de la criptografía de clave pública, debe verificarse siempre en el certificado que la clave siga siendo válida.

d. Retirada.
Si se sospecha, por cualquier motivo, que la clave ha sido comprometida, ha de acudirse a la Autoridad de Certificación para comunicárselo y que ésta proceda a certificar una nueva clave.

e. Terminación.
Una vez que la clave finaliza su ciclo de vida, se almacena y es reemplazada por una nueva.

5.4.1. ALMACENAMIENTO Y GESTIÓN DE LAS CLAVES.

Uno de los problemas principales que aparece a la hora de utilizar ***Criptografía*** es el de almacenamiento de las claves. El grado de seguridad con el que se almacena una clave debe ser directamente proporcional a la importancia de los mensajes que deben ser cifrados con dichas claves. Un método idóneo puede ser el uso de tarjetas inteligentes, que acceden al sistema mediante el *hardware* adecuado.

5.4.2. RECUPERACIÓN DE CLAVES (*KEY RECOVERY*).

Uno de los argumentos del gobierno de los EE.UU. para impedir la exportación de productos criptográficos "fuertes", es la posibilidad de que éstos sean utilizados por gobiernos enemigos, terroristas o criminales en general, con lo que se vería amenazada la seguridad nacional. Sin embargo, la presión de la industria informática de los EE.UU. es fuerte y se vislumbra una relajación de las restricciones. La contrapartida es la introducción de mecanismos de recuperación de claves que permitan, bajo estricto mandato judicial, levantar las protecciones criptográficas de comunicaciones determinadas.

Con este trasfondo, y no ligadas específicamente a la política del gobierno de EE.UU., se han desarrollado dos técnicas, conceptualmente muy similares, para asegurar la gestión y almacenamiento de las claves (y eventualmente, su recuperación):

1. ***Key Escrow*** **(custodia de claves):** es el usuario u organización quien genera su clave o claves y las entrega a otra parte que la guarda para él. Variantes de este mecanismo consisten en fragmentar la clave y confiar cada fragmento a custodios diferentes. De este modo, la protección de la clave queda en otras manos.

2. ***Trusted Third-Party*** **(tercera parte de confianza):** en este caso, es una tercera parte la que genera la clave correspondiente a requerimiento del usuario, la distribuye a los receptores correctos y almacena una copia para sí misma.

   La seguridad de la clave queda de nuevo en otras manos diferentes a las de los usuarios.

En conclusión podemos decir que el papel fundamental de una **Autoridad Certificadora** es: “Generar y firmar los **Certificados Digitales** de los usuarios y mantener el status correcto de los certificados”.

Capítulo 6

OTRAS HERRAMIENTAS CRIPTOGRAFICAS

6.1. INTRODUCCIÓN.

En esta parte del manual nos dedicaremos exclusivamente a enumerar otros tipos de herramientas o técnicas que son usadas en el ámbito de la ***Criptografía Moderna***, donde cada una de ellas tiene una gran aplicación y un propósito muy específico. Sin embargo su descripción y funcionamiento detallado no es el propósito de este manual ni de esta sección, así que solo se mencionarán algunos aspectos de importancia.

6.2. COMPARTICIÓN DE SECRETO.

La ***Compartición de Secreto***, como su nombre lo dice es una técnica criptográfica que se dedica a partir un secreto, que puede ser una clave secreta, en la responsabilidad de varias personas y que solo con el numero mínimo de personas se podrá reconstruir el secreto compartido.
Por ejemplo: Si el secreto es el número 100 y este debe ser compartido por tres personas **A1, A2, y A3**; una forma de poder hacerlo es generar un numero aleatorio menor a 100, digamos el **33** posteriormente se genera otro numero aleatorio menor a 100-33, digamos el **67**, y finalmente la tercera parte será 100-(21+33)=**46**. Así el secreto 100 esta compartido por **A1(33), A2(67), y A3(46)** cada cual con su parte correspondiente. Como podemos observar ninguno conoce las partes de los demás. Conste que esto es solo para explicar el concepto.

La ***Compartición de Secretos*** puede ser usada para compartir, digamos la combinación de una caja fuerte, la clave de lanzamiento de algún proyectil, la clave secreta de una Autoridad Certificadora, la clave de activación de algún dispositivo de alto riesgo y otras actividades que tengan que ver con una fuerte seguridad.

6.2.1. ESQUEMA "LÍMITE DE SHAMIR".

Uno de los mejores métodos de ***Compartición de Secretos*** y más conocido es el esquema (n,k) ***Limite de Shamir.*** Este método consiste en partir una clave ***k*** en ***n*** partes, y se tiene como mínimo (limite) el numero ***k*** de partes para reconstruir la clave, es decir cualquiera ***k*** de los ***n*** custodios pueden reconstruir la clave ***k***, pero ningún subgrupo de ***k-1*** custodios podrá hacerlo.

El esquema ***Límite de Shamir*** se basa en lo siguiente:

1. Se define el número de custodios **t**, digamos **t=2**.
2. Se generan aleatoriamente los coeficientes necesarios para construir un polinomio de **t-1** grado, en nuestro caso: $\mathbf{f(x)=s+a_1x}$ $\mathbf{s_1=f(1).}$
   Donde el coeficiente es aleatorio y **s** es el secreto a compartir.
3. Evidentemente el secreto se recupera conociendo el polinomio y evaluando en cero **s=f(0)**.
4. Para nuestro caso las partes serán:
   $s_2=f(2)$
   $s_2=s+2a_1$
   $s_i=f(i)$
   $s_1=s+a_1$

Por tanto el método para recuperar el secreto **s**, es reconstruir el polinomio **f(x)** a partir de **t** partes cualquiera, esto se hace por medio **d** la Interpolación de Lagrange.

6.3. CRIPTOGRAFÍA VISUAL.

Una idea ingeniosa de usar un método de comparación de secretos con un esquema limite (n,k) es la ***Criptografia Visual,*** la cual consiste en lo siguiente: Una imagen es partida en **n** partes, y si se sobreponen al menos **k** de estas partes se puede reconstruir la imagen.
Por ejemplo: De un esquema limite, trabaja considerando que si la imagen es de blanco y negro, entonces la imagen podrá ser un conjunto de cuadros completamente negros, por ejemplo la siguiente imagen:

Figura 6.1

Ahora cada cuadro de la imagen podrá ser considerado como blanco y negro, equivalentemente con valores 0 y 1, para partir esta imagen en dos partes n=2 y considerando el límite k=2, se produce como sigue:

Ψ Cada cuadro que es completamente negro podrá ser partido en dos partes de la siguiente forma:

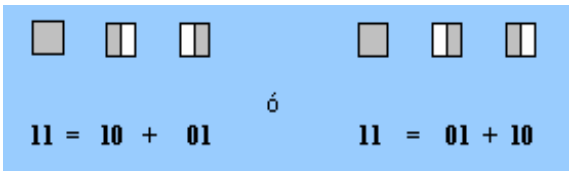


Figura 6.2

Ψ Y cuando completamente esté blanco podrá ser partido en dos de la siguiente manera:

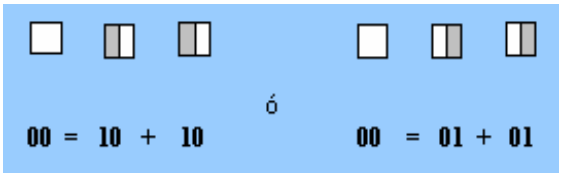


Figura 6.3

La suma modulo 2 significa: 1+0=1, 0+1=1, 0+0=0; pero también 1+1=0, de este modo se puede tomar cualquiera de las dos particiones de los cuadros de color blanco.

Ψ Para formar las dos partes de la figura se elige aleatoriamente una de las combinaciones anteriores según se parta un cuadro blanco o uno negro.
En el caso de la figura anterior (llave), una vez elegida las partes, la figura partida en un esquema limite queda así:

Parte 1. *Parte 2.*

De esta forma se tiene partida la figura en dos partes y se recuperara solo sobreponiendo una sobre la otra.

Ψ Al sobre poner las dos partes se recupera la figura, de la siguiente forma:

Figura 6.5

En el caso general se parte los cuadros blancos y negros en ***n*** pedazos y hasta no tener ***k*** pedazos negros el cuadro reconstruido será siendo blanco, a partir de ***k*** pedazos negros hasta ***n*** el cuadro reconstruido será negro. En nuestro caso, un cuadro con solo la mitad negra será considerado blanco, es necesario que tenga dos mitades negras par que el cuadro reconstruido se considere negro, que es el caso del esquema anterior.

6.4. DINERO ELECTRÓNICO.

Otra de las aplicaciones de gran importancia que puede ser realidad gracias a la ***Criptografía de Clave Pública*** es conocida como ***Dinero Electrónico.***
Esta aplicación, en términos sencillos es otra representación de lo que conocemos como dinero o valor.
Por ejemplo: Tenemos dinero en billetes emitidos por algún país, podemos tener cheques pagaderos en un banco, bonos, y muchas otras formas de representar al dinero.

El ***Dinero Electrónico,*** es físicamente un número que se genera aleatoriamente y se le asigna un valor, se cifra, se firma y se envía al banco, para luego el banco validar el número y certificar el valor. Posteriormente éste lo regresa firmado al usuario para que el usuario pueda efectuar alguna transacción con ese billete electrónico.

= f5 6ª 80 b2.....03

Figura 6.6

6.4.1. PROPIEDADES DEL DINERO ELECTRÓNICO.

- **Independencia:** La seguridad del dinero digital no debe depender del lugar físico donde se encuentre.
  Ej: Disco duro de una máquina.

- **Seguridad:** El dinero digital (el número), no debe de ser usado en dos diferentes transacciones.

- **Privacidad:** El dinero electrónico debe de proteger la privacidad de su usuario, de ésta forma cuando se haga una transacción debe de poder cambiarse el número a otro usuario sin que el banco sepa que dueños tuvo antes.

- **Pagos fuera de Línea:** El dinero electrónico no debe de depender de la conexión de la red, así un usuario puede transferir dinero electrónico que tenga en una "Smart Card" a una computadora, el dinero digital debe ser independiente al medio de transporte que use.

- **Transferibilidad:** El dinero electrónico debe de ser transferible. Cuando un usuario transfiere dinero electrónico a otro usuario debe de borrarse la identidad del primero.

- **Divisibilidad:** El dinero electrónico debe de poder dividirse en valores fraccionarios según sea el uso que se da.
  Ej: En valor de 100, 50 y 25.

Veamos ahora los pasos que se siguen para realizar una transacción con dinero electrónico a través de un ejemplo:

1. A *genera un número aleatorio grande* N *de digamos* 100 dígitos *y le da un valor digamos* 1000 pesos.
2. A *cifra este número junto a su valor con su clave secreta asimétrica.*
3. A *firma este número y lo transmite a su banco.*
4. *El banco de* A *usa, la clave pública de* A *para descifrar el número y verificar la firma de* A *del documento electrónico*
5. *El banco revisa que* A *tenga en sus cuentas la cantidad pedida,* 1000 pesos *y la debita de alguna de estas cuentas.*
6. *El banco firma el número que mando* A, *con el valor asignado de* 1000 pesos.
7. *El banco regresa el número que ya es dinero hacia* A.
8. A *envía este dinero a* B.
9. B *verifica la firma del banco de* A *que está en* N.
10. B *envía* N *a su banco.*
11. *El banco de* B *no verifica la firma del banco de* A *en* N.

12. *El banco de* B *verifica que* N *no este en la lista de números (ya usados).*
13. *El banco de* B *acredita la cantidad de* 1000 pesos *a la cuenta de* B.
14. *El banco de* B *pone a* N *en la lista de números (ya usados).*
15. *Finalmente el banco de* B *envía un recibo firmado donde establece que tiene* 1000 pesos *mas en su cuenta.*

6.5. COMERCIO ELECTRÓNICO.

Como hemos visto, hoy en día gran parte de la actividad comercial ha podido transformarse gracias a las redes mundiales de conexión por computadora como es el caso de Internet, ésta transformación facilita hacer transacciones en cualquier momento de cualquier lugar del mundo.
Todo lo que está alrededor de ésta nueva forma de hacer negocios es lo que le llamaremos ***Comercio Electrónico.*** Sin duda la gran variedad de actividades que giraban alrededor del que hacer comercial se han tenido que unir con las nuevas técnicas cibernéticas. Así hoy tanto un comerciante, un banquero, un abogado o un matemático puede hablar de ***Comercio Electrónico*** enfocándose a la parte que le corresponde.

Existen diferentes niveles de hacer ***Comercio Electrónico***, y su clasificación aún está por formarse, sin embargo, la parte más visible es la que cualquier usuario en una computadora personal puede ver. Esto de hacer ***Comercio Electrónico*** se convierte a comprar o vender usando una conexión por Internet en lugar de ir a una tienda.

La forma de hacer esto es muy similar a lo que normalmente se hace:
Por ejemplo: En la tienda uno entra al establecimiento, de forma electrónica se prende la computadora y una vez conectado a Internet se entra a la página del negocio. Enseguida un comprador revisa los productos que posiblemente compre y los coloca en un carrito, de la misma forma en la computadora se navega por la página del negocio y con el browser se revisan los productos que este vende, al escoger estos se colocan en un carrito virtual, que no es más que un archivo de usuario.
Una vez elegido bien los productos de compra se pasa a la caja, donde se elige un sistema de pago y se facturan los productos al comprador. De forma similar en la computadora se pueden borrar productos que no se quieren comprar o agregar nuevos. Una vez elegidos estos se procede a una parte de la página que toma los datos y solicita el método de pago, generalmente se lleva a cabo con tarjeta de crédito.

En la parte tradicional de comprar al pagar en la caja termina el proceso, en la parte por computadora aún tiene que esperarse que sean enviados los productos comprados.

A pesar de esto las ventajas que ofrece el ***Comercio Electrónico*** son magníficas, ya que es posible comprar en un relativo tiempo corto una gran cantidad de productos sin necesidad de moverse de lugar, es decir al mismo tiempo se puede comprar una computadora, un libro, un regalo, una pizza, hacer una transacción bancaria; de la forma tradicional se llevaría al menos un día completo y eso si los negocios están en al misma ciudad, si no, el ahorro de tiempo que representa comprar por Internet es incalculable.

Al efectuar una operación comercial por Internet se presentan nuevos problemas:

Por ejemplo: ¿Cómo saber que la tienda virtual existe verdaderamente?. ¿Una vez hecho el pedido, como saber que no se cambia la información?. ¿Cuándo se envía el número de tarjeta de crédito, como saber si este permanecerá privado?.
Para el comerciante también se le presentan algunos problemas similares:
Por ejemplo: ¿Cómo saber que el cliente es honesto y no envía información falsa?.

Como vemos todos estos problemas pueden ser resueltos de manera satisfactoria si se implementan protocolos de comunicación segura usando ***Criptografía*** (como es el caso de implementar protocolos vistos en el *Capitulo 4: Aplicaciones de Seguridad)*.

Para finalizar veamos a continuación como se desarrollaría un caso práctico de ***Comercio Electrónico:***

1. Un cliente, que tiene una tarjeta (supongamos que se llama Terry) accede al comercio y navega por él, seleccionando una serie de productos.
2. Terry rellena una orden de compra, calculándose el monto total de la operación (incluyendo los gastos de envío).
3. Terry selecciona el método de pago (tarjetas *Visa*, *MasterCard*, *American Express*).
4. Terry envía su orden de pago con el método de pago elegido al comercio.
5. El comercio solicita autorización por parte del banco de Terry para llevar a cabo la transacción.
6. El comercio le envía a Terry la confirmación de la compra (en forma de factura pro-forma por ejemplo).
7. El comercio envía los bienes comprados a Terry.
8. El comercio solicita a su banco el abono de los bienes comprados.

CONCLUSION.

Es importante resaltar que el elemento fundamental de un **Sistema Informático** es la **SEGURIDAD**, y como tal, ésta debe proporcionar los servicios para que un

sistema o aplicación trabaje de manera fiable. No debemos confiar ni menos descuidar el acceso de extraños o potenciales atacantes a nuestros PC's ya que se podrían producir pérdidas de datos y no poder recuperarlos.

Pero la **SEGURIDAD** no sólo se consigue con la ***Criptografía***, u otras herramientas de seguridad, además es necesario dotar al sistema de razones que le demuestren que se va a hacer buen uso de los datos que manipulará; por ejemplo un negocio o aplicación que funcione a través de la red, el cual debe ser, lo más transparente posible, estimulando a los usuarios a que realicen las operaciones con una total confianza y tranquilidad.

Cabe entonces destacar y entender que la ***Criptografía*** es un subconjunto de algo mucho mayor como es la **Seguridad Informática**. Aunque la ***Criptografía*** ofrece un amplio margen de seguridad en un **Sistema Informático** y es una de las herramientas más importantes de la **SEGURIDAD,** no quiere decir que alguien no pueda tener el acceso a un archivo encriptado y que pueda borrarlo, ya que se trata simplemente de un archivo.

RECOMENDACION.

Determinadas empresas que cuenten con un sistema computarizado, y que a su vez alberga información confidencial deberá hacer uso de los sistemas criptográficos para la protección de sus datos.

Que nuevas generaciones que opten por la investigación de este tema, den un aporte mas con nuevos algorítmos criptográficos.

Capítulo 7

USO DE LA APLICACIÓN

7.1. INTRODUCCIÒN.

Esta aplicación llamada **"CriptoSistemas Modernos de Clave Pública y Privada",** fue realizada para un Proyecto de Desarrollo de Software de práctica de la asignatura de Seguridad Informática por los alumnos egresados de la Universidad Politécnica de Madrid (U.P.M) D. José Alberto Charfolé y D. Abel Gregorio Palomino, cuyo tutor fue el D. Jorge Ramió Aguirre quien nos facilitó esta aplicación para demostrar de manera práctica como funcionan algunos de los Algorítmos de encriptación que describimos teóricamente en capítulos anteriores en este manual.

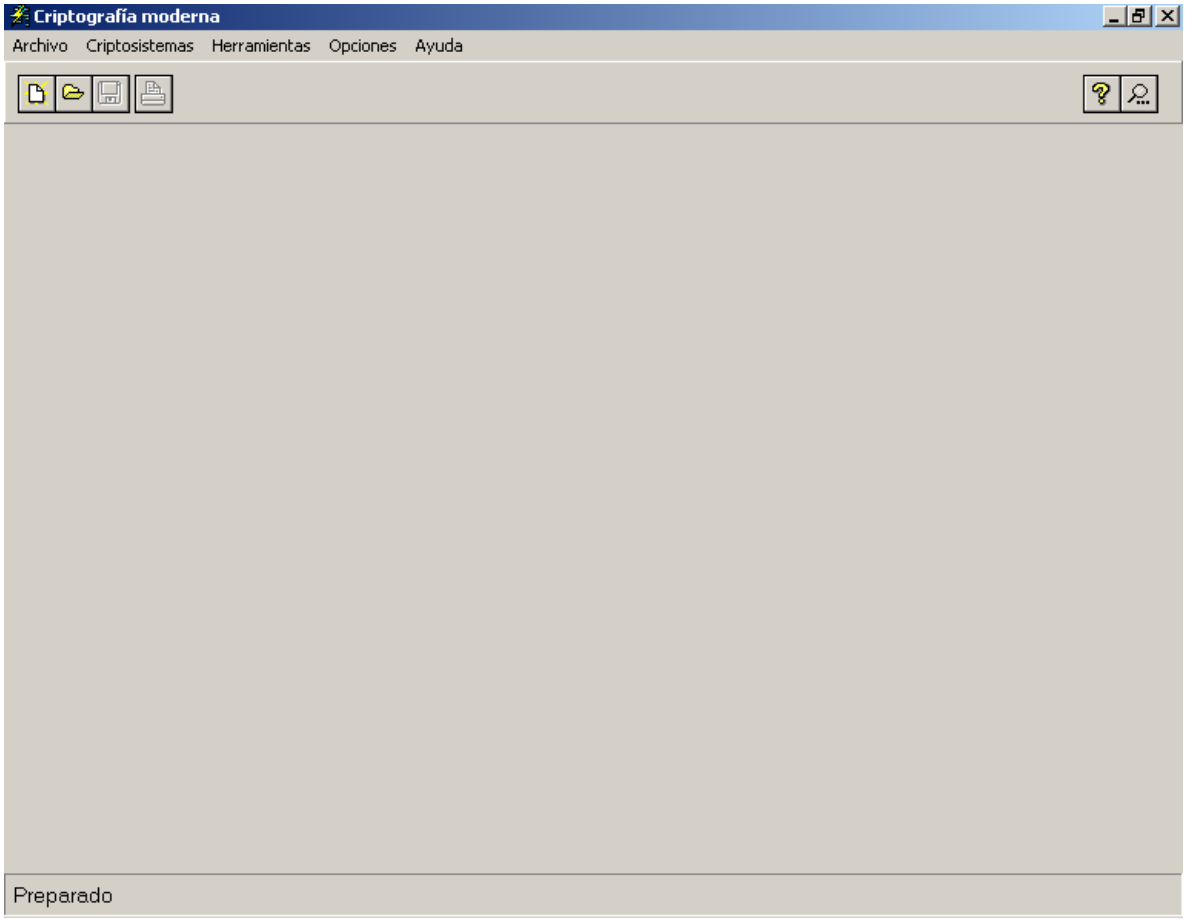
Recurrimos al profesor Ramió por ser gran conocedor de la rama o materia, ya que en un principio el objetivo de nosotros era crear una aplicación con similitud a esta, pero se nos presentó el problema de los algorítmos de encriptación; los cuales ya están programados solo para venderse (por la empresa que posee la patente del mismo), e implementarse en la aplicación que se vaya a utilizar y no contamos con los recursos para realizar dicha acción, así que el profesor Ramió nos facilitó directamente esta aplicación denominada CriptoSistemas Modernos, sin ningún tipo de problema, siempre y cuando sea utilizada para fines académicos.

7.2. DESCRIPCIÓN DE LA APLICACIÓN.

El propósito de la aplicación **"CriptoSistemas Modernos",** es demostrar como funciona un CriptoSistema Moderno; básicamente encritpar / desencriptar utilizando algorítmos de encriptación moderna.

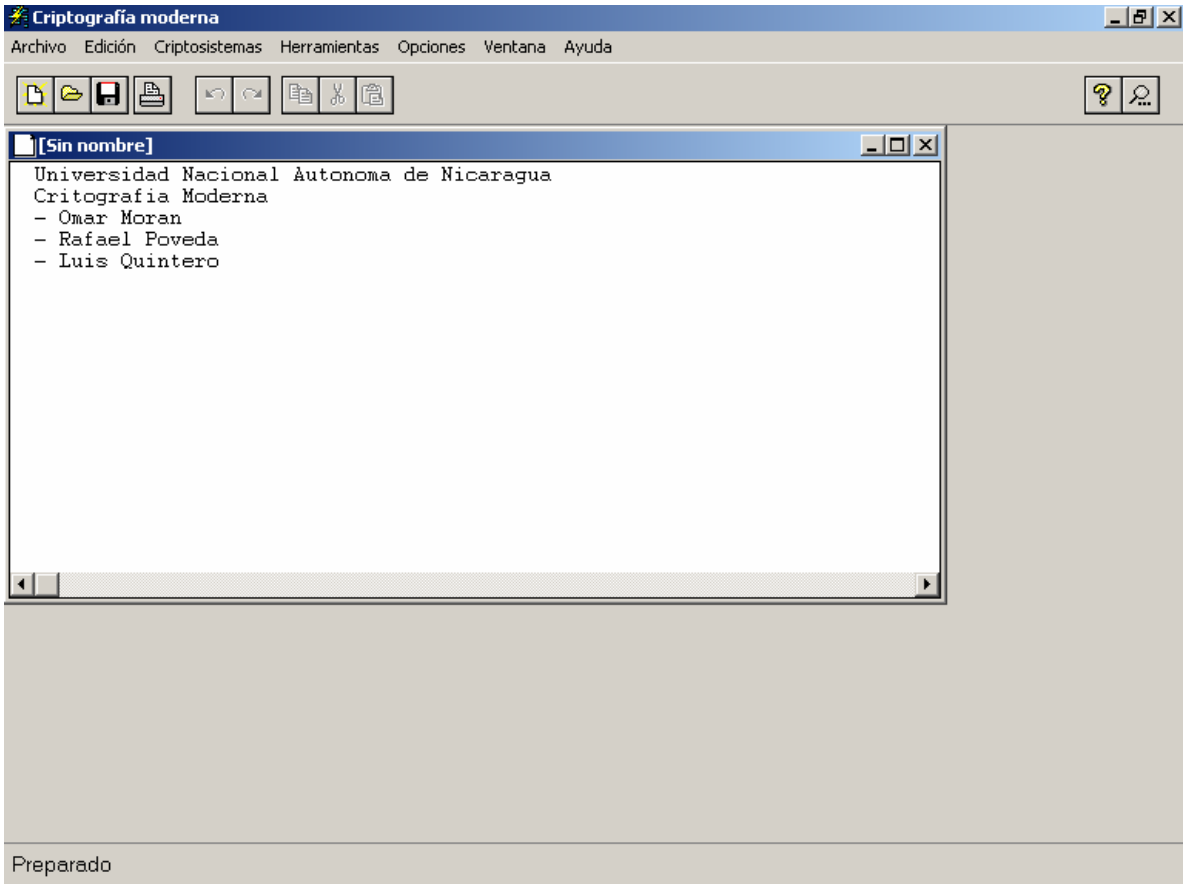
En cuanto a la instalación se refiere, puede hacerse en cualquier plataforma de Windows (95, 98, NT, 2000, XP), la aplicación es fácilmente portable, su tamaño es de 1.54 Mb y una vez instalado en disco duro ocupa 3.08 Mb.

Consta como lo es en la mayoría de las aplicaciones con una pantalla inicial presentando el nombre del software y las personas por las que fue realizado y dirigido, seguidamente se presenta la pantalla principal del software que contiene lo siguiente: Presenta una barra de menús que consta de varias opciones como lo son: ***Archivo, CriptoSistemas, Herramientas, Opciones, Ayuda*** y por la parte de abajo una barra de estado indicando que se encuentra "Preparado" par ejecutar alguna acción. Gráficamente la interfaz se vería así:



.

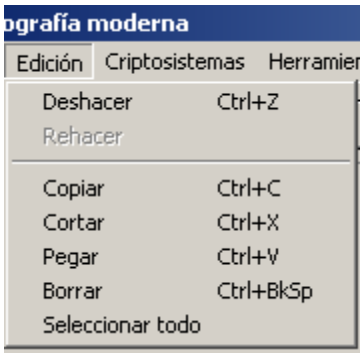
Este software trabaja con lo que llamamos M.D.I (Aplicación Multidocumento), donde podremos crear y editar ficheros de texto, además servirá de entrada de los distintos algorítmos para su encriptación. En consecuencia la pantalla que visualizaremos para ver la introducción de una cadena de caracteres o texto será la siguiente:



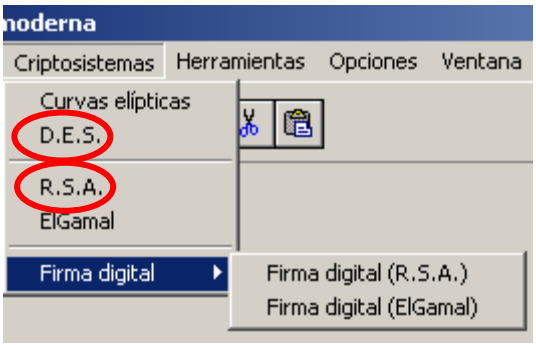
Así mismo podremos introducir cualquier cantidad de texto sin limite teórico, para que luego sea guardado, y a continuación el usuario escoja alguna opción y pueda trabajar con dicho fichero.

En cuanto al menú ***Archivo*** contiene las siguientes opciones para poder manipular y dar mantenimiento a todas los archivos de texto creados en el editor de texto.

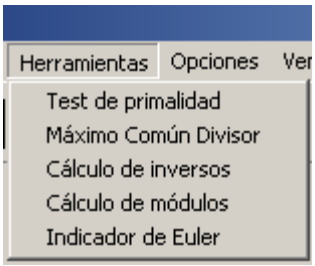
Menú ***Edición,*** tiene las opciones necesarias para cambiar o manipular la cadena o texto introducido, estas opciones son:



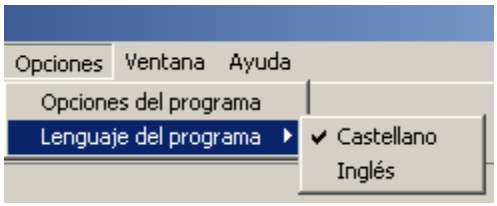
El menú ***Criptosistemas*** por su parte consta de los algorítmos que se encuentran en esta aplicación para realizar las operaciones de encriptación y desencriptación, Los algorítmos en los cuales haremos mayor hincapié serán los siguientes: **D.E.S** (Clave Privada), **R.S.A** (Clave Publica) por ser los mas conocidos y usados en cuanto al tipo de CriptoSistema se refiere:



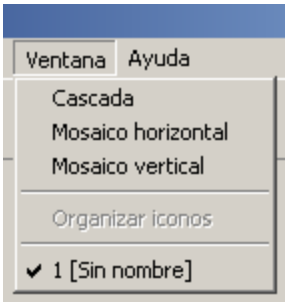
Por su parte en el menú ***Herramientas***, se incluyen diferentes opciones para facilitar y ayudar de alguna manera al usuario a la acción que él decida (encriptar/ desencriptar), aplicando estas herramientas cuando sea necesario utilizar, ya que cada algorítmo tiene su forma de utilizar y herramienta que usar.



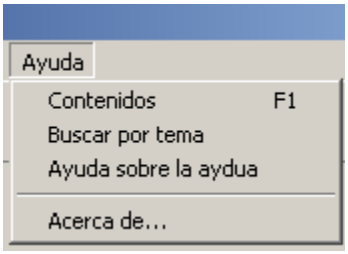
En el menú ***Opciones*** se determinan las características generales del software como: el lenguaje del mismo, el editor de texto por defecto, la extensión de los archivos o ficheros de texto, extensión de los archivos a cifrar o descifrar etc.



En el Menú ***Ventana*** se encuentran las opciones comunes para trabajar o controlar el uso de ventanas.



Y por último se encuentra el Menú ***Ayuda*** (para su uso se debe de tener como conocimientos mínimos lo que es el concepto de aplicación M.D.I, manejo de ventanas y menús); que consta de una extensa y completa ayuda y guía del uso y funcionamiento de la aplicación, además aporta descripciones teóricas de los métodos criptográficos y herramientas utilizadas en este el software.

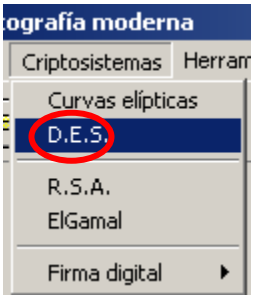


7.3. USO DE D.E.S EN LA APLICACIÓN (CriptoSistemas Modernos).

7.3.1. ENCRIPTAR / DESENCRIPTAR CON D.E.S.

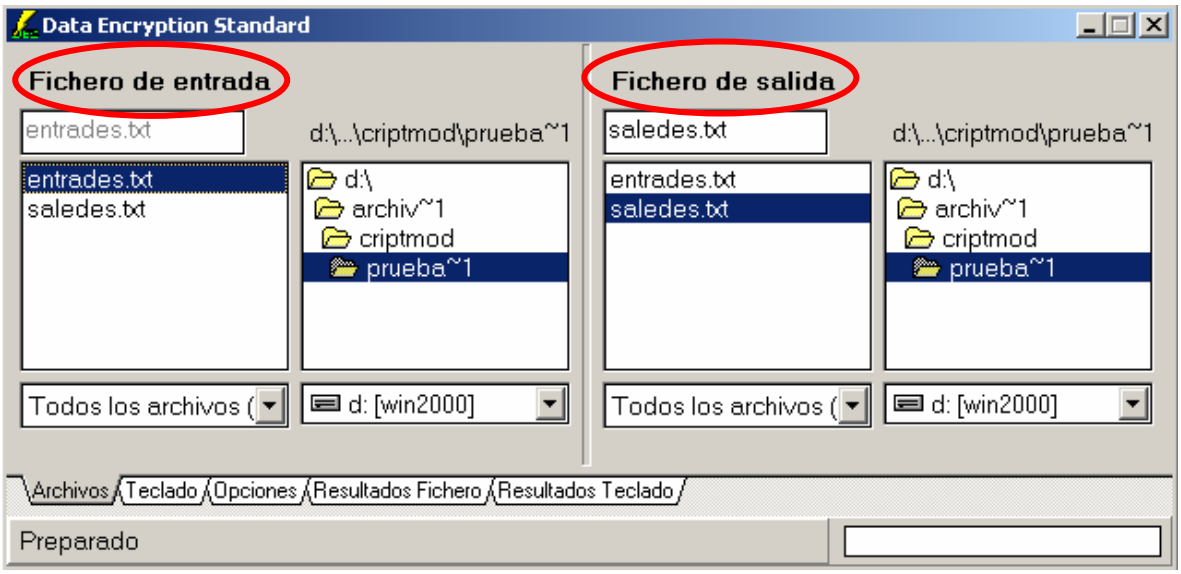
Para realizar el proceso de encriptación con el algoritmo de Clave Privada D.E.S, utilizando la aplicación Criptosistemas Modernos básicamente debemos seguir los pasos siguientes:

1. Debemos de escoger el Algorítmo de Encriptación D.E.S, que se presenta en el menú **Criptosistemas – D.E.S**. Visto gráficamente se vería así:

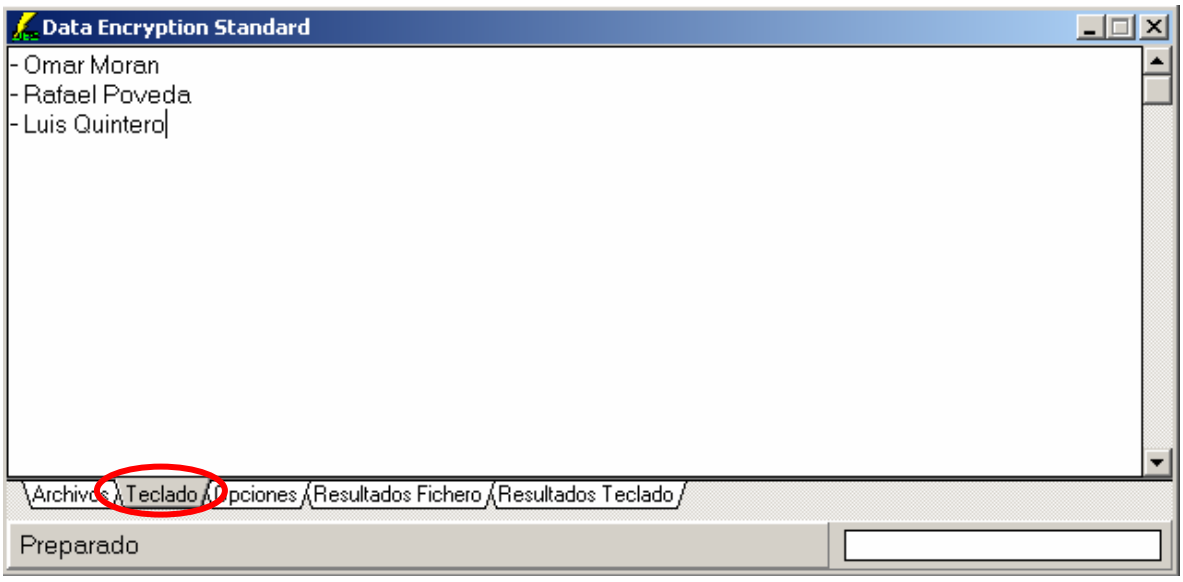


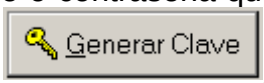
2. Luego debemos escoger la información de entrada con la que se va a trabajar, ya sea para **Cifrar o Descifrar** la información, la cual se puede presentar de dos maneras:

- Seleccionando un archivo o **fichero de entrada** creado con anterioridad en el editor de texto de la aplicación (u otro editor de texto), que servirá a D.E.S para trabajar y tomar como fuente de información, ya sea para cifrar o descifrar, y otro **fichero de salida** que será donde se obtendrá el resultado de la operación realizada con el fichero de entrada.

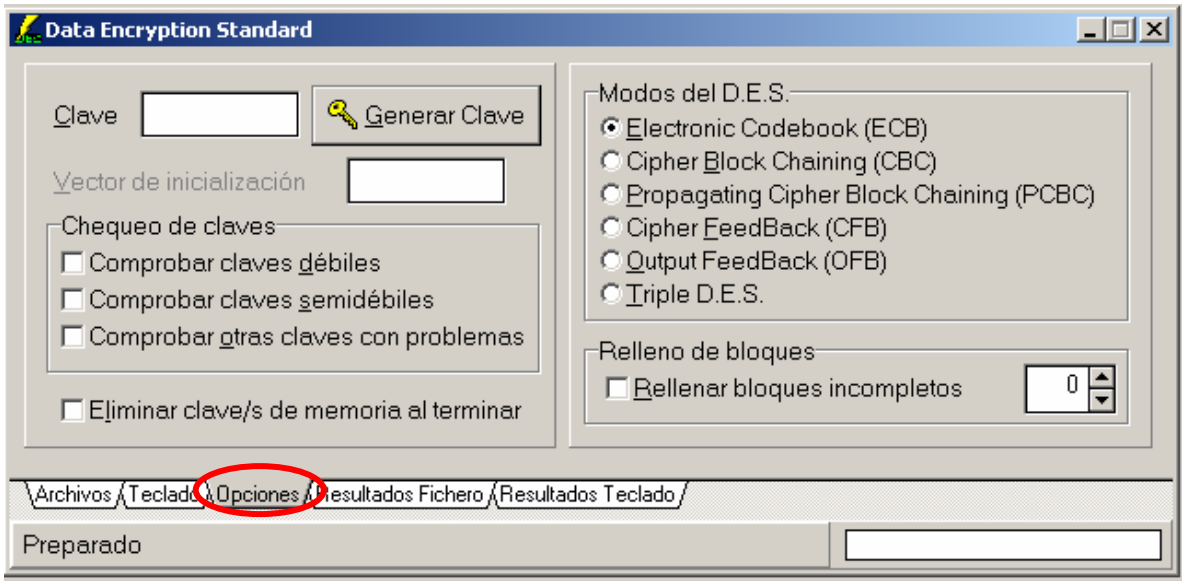


- La otra manera es, introducir la cadena de caracteres o texto es pulsando directamente la pestaña **teclado.** La cadena o texto a introducir debe tener un mínimo de ocho caracteres. Esta opción se recomienda para realizar pruebas sobre pequeña cantidad de información donde su resultado no tiene mucha importancia o no se interesa conservar. La interfaz que presenta es la siguiente:

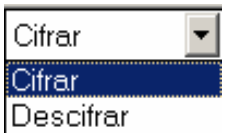


3. A continuación elegimos la pestaña **Opciones** donde es necesario que se introduzca una clave o contraseña que puede ser generada aleatoriamente con el botón de comando 

, o introducida manualmente con un mínimo de 8 (ocho caracteres), que pueden ser símbolos, números y letras. En este formulario además podemos encontrar otras opciones de la manera como opera D.E.S, opciones que se le pueden atribuir a una clave o contraseña y la implementación del algoritmo **T.D.E.S**.

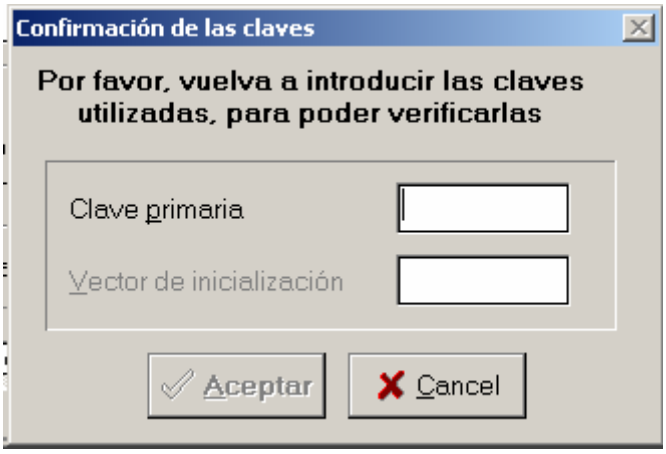


4. Luego escogemos la operación que queremos realizar sobre dicho fichero (en caso de que exista), o sobre el texto o cadena introducida por teclado; estas son dos: **Cifrar** y **Descifrar.** Gráficamente la interfaz de estas opciones se vería así:

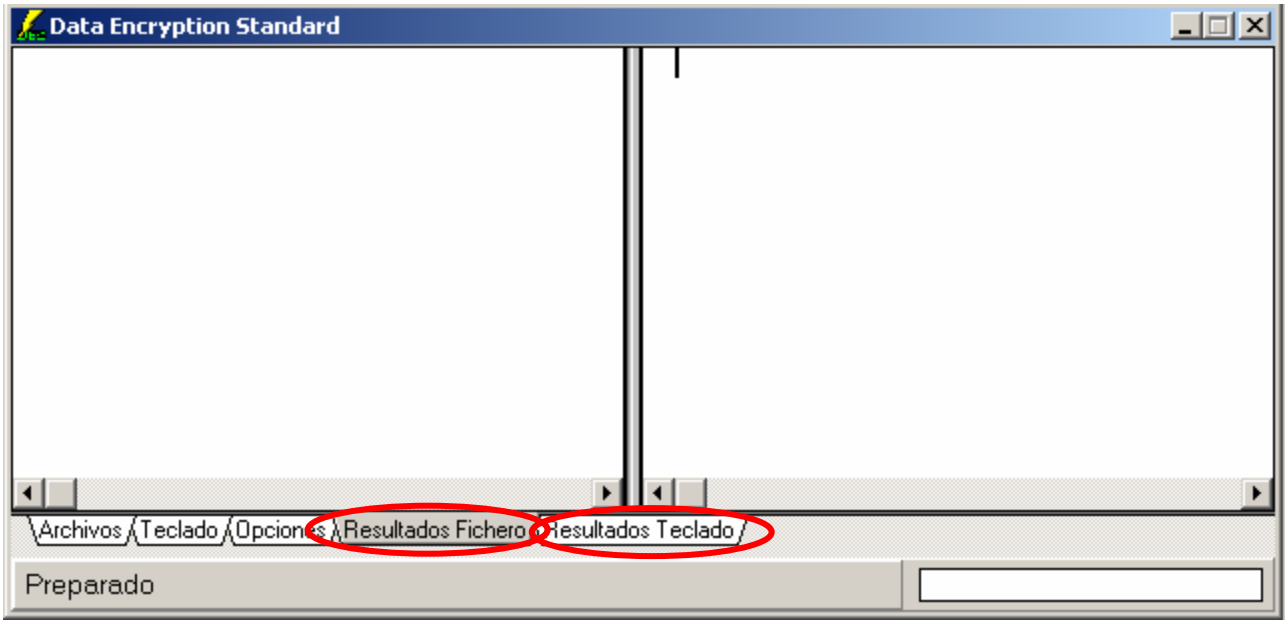


5. Seguidamente pulsamos el botón de comenzar que tiene el siguiente aspecto:

6. Nos mandara un mensaje en una caja de dialogo donde nos pedirá que ingresemos de nuevo la clave utilizada para que pueda ser verificada por el sistema y ejecute la acción.



7. Por ultimo, para visualizar el resultado de la operación realizada, pulsamos la pestaña ***Resultados Fichero*** en caso de que se haya trabajado con ficheros de lo contrario pulsamos la pestaña ***Resultados Teclado*** en caso de que hayamos introducido la cadena por teclado. Gráficamente estos dos formularios tienen una interfaz similar, esta es:

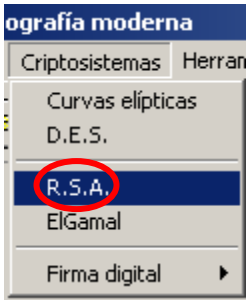


7.4. USO DE R.S.A EN LA APLICACIÓN (CriptoSistemas Modernos).

7.4.1. ENCRIPTAR / DESENCRIPTAR CON R.S.A.

Para realizar el proceso de encriptación con el algoritmo de Clave Privada R.S.A, utilizando la aplicación Criptosistemas Modernos básicamente debemos seguir los pasos siguientes:

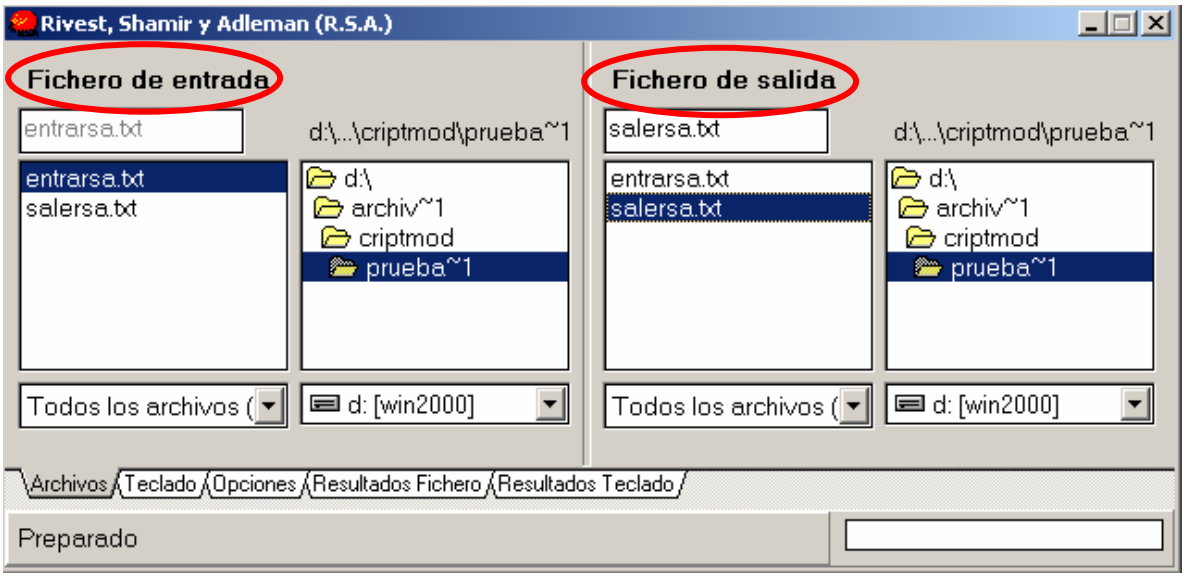
1. Debemos de escoger el Algoritmo de Encriptación R.S.A, que se presenta en el menú **Criptosistemas – R.S.A**. Visto gráficamente se vería así:



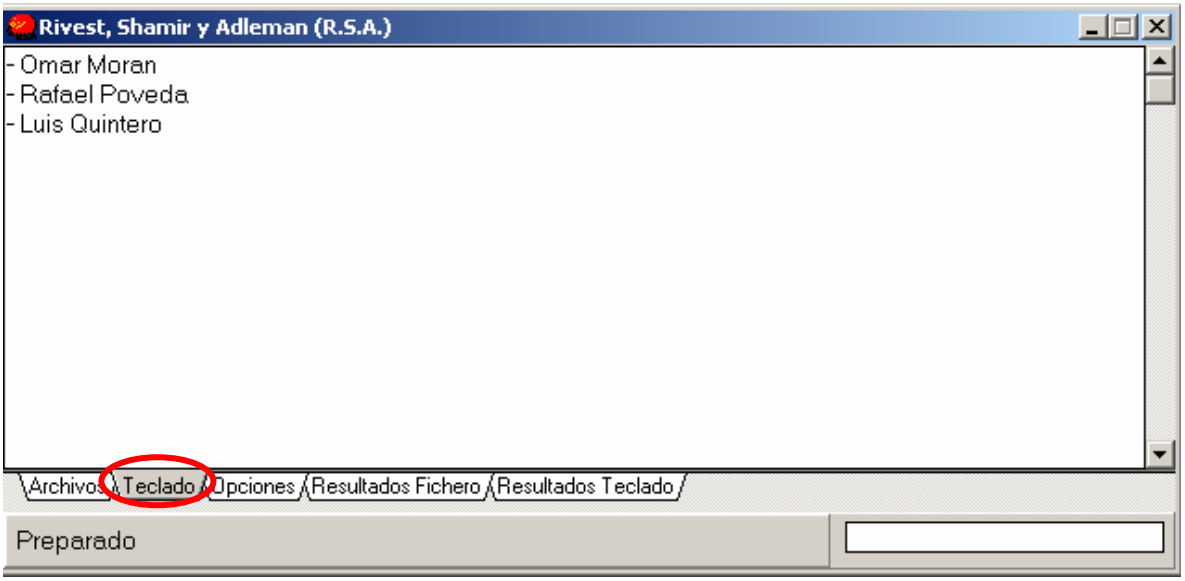
2. Luego debemos escoger la información de entrada con la que se va a trabajar, ya sea para **Cifrar o Descifrar** la información, la cual se puede presentar de dos maneras:

- Seleccionando un archivo o **fichero de entrada** creado con anterioridad en el editor de texto de la aplicación (u otro editor de texto), que servirá a R.S.A para trabajar y tomar como fuente de

información, ya sea para cifrar o descifrar, y otro **fichero de salida** que será donde se obtendrá el resultado de la operación realizada con el fichero de entrada.

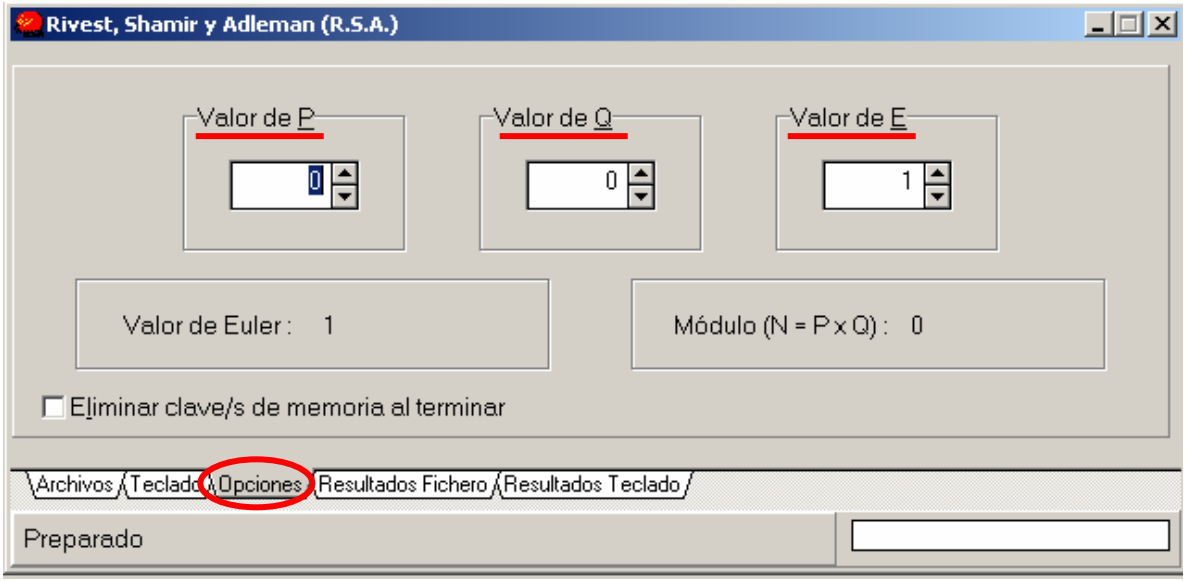


- La otra manera es, introducir la cadena de caracteres o texto es pulsando directamente la pestaña **teclado.** La cadena o texto a introducir debe tener un mínimo de ocho caracteres. Esta opción se recomienda para realizar pruebas sobre pequeña cantidad de información donde su resultado no tiene mucha importancia o no se interesa conservar. La interfaz que presenta es la siguiente:



3. Luego elegimos la pestaña **Opciones**, donde debemos de introducir lo siguiente: ***Valor de P*****,** es un valor que junto al producto de Q forman el modulo de trabajo N, P debe estar comprendido en el rango de 0 – 46.340 y debe ser un número primo. ***Valor de Q*****,** es un valor que junto al producto de P forman el modulo de trabajo N, Q debe estar comprendido en el rango de 0 – 46.340 y debe ser un número primo. ***Valor de E*****,** este numero debe ser positivo, su rango debe de estar entre 1 – 46.340, además debe cumplir los requisitos siguientes:

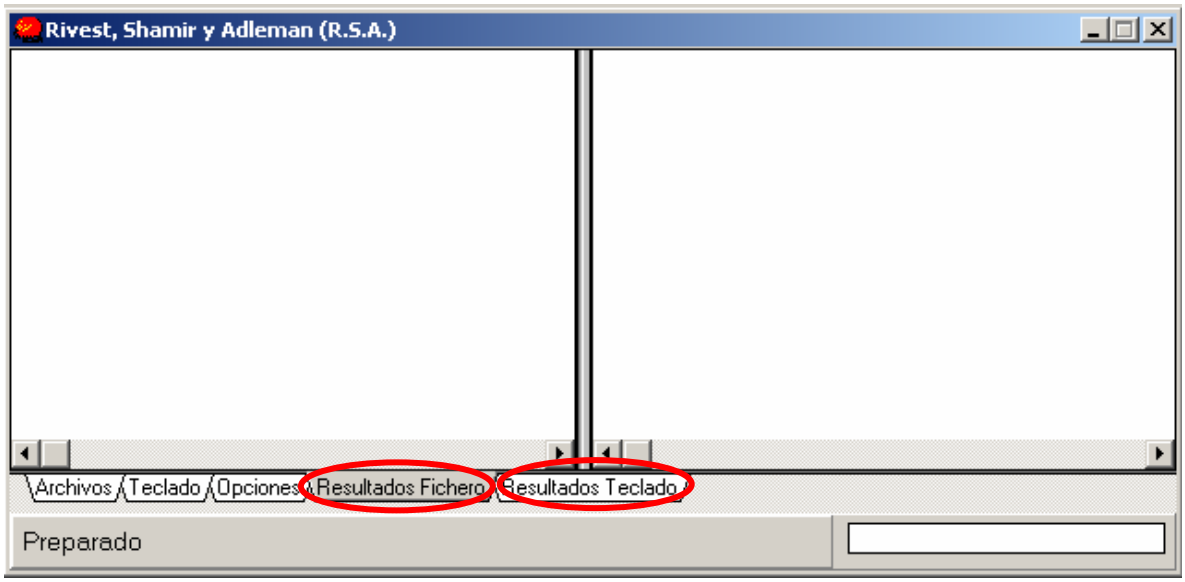
- El rango debe ser $1 \leq e \leq N$.
- Debe de ser primo con el valor de Euler de N.



4. Luego escogemos la operación que queremos realizar sobre dicho fichero (en caso de que exista), o sobre el texto o cadena introducida por teclado; estas son dos: **Cifrar** y **Descifrar.** Gráficamente la interfaz de estas opciones se vería así:

5. Seguidamente pulsamos el botón de comenzar que tiene el siguiente aspecto:

6. Por ultimo, para visualizar el resultado de la operación realizada, pulsamos la pestaña ***Resultados Fichero*** en caso de que se haya trabajado con ficheros de lo contrario pulsamos la pestaña ***Resultados Teclado*** en caso de que hayamos introducido la cadena por teclado. Gráficamente estos dos formularios tienen una interfaz similar, esta es:


Rivest, Shamir y Adleman (R.S.A.)
Archivos
Teclado
Opciones
Resultados Fichero
Resultados Teclado
Preparado

GLOSARIO

A

Algorítmo: Secuencia finita y ordenada de instrucciones elementales que, dados los valores de entrada de un problema, en algún momento finaliza y devuelve la solución.

Acceso: Acción de llegar o acercarse a algo. Entrar o dar paso a determinada información.

Algorítmo de Encriptación: Sistema de pasos empleados para cifrar o encriptar un texto en claro.

Amenaza: Dar a entender con actos que se quiere causar un mal a algún tipo de información.

Análisis: Distinción y separación de las partes de un todo hasta llegar a conocer sus principios o elementos

Análisis de Riesgo: Consiste en enumerar todos los riesgos a los cuales esta expuesta la información

Aplicación: Programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo

Archivo: Dicese así a los elementos que se guardan en un computador. Al conjunto de estos elementos se les denomina información.

Archivo Cifrado: Información que no puede ser leída por personas que no tienen los permisos para hacerlo

Atacante: Elemento que puede atentar contra la seguridad de un Sistema computarizado, intentando dañar el sistema.

Ataque Informatico: Ataque a un sistema computarizado.

B

Backdoor: Puerta trasera, denominada así al procedimiento para poder intervenir el mensaje cuando el diseñador del programa encriptador así lo desee. Método para recuperar un mensaje en claro a partir del mensaje cifrado sin utilizar búsquedas exhaustivas de la clave.

Bombas Lógicas: Parte del código de un programa que permanece sin hacer nada útil en el computador hasta que son activadas, con la finalidad de realizar alguna acción perjudicial al sistema computarizado.

Bugs: Errores producidos de manera involuntaria por los programadores de sistemas o aplicaciones, aun asi siendo peligrosos para un sistema.

C

Cadena (String): Grupo de caracteres o bytes de caracteres que se procesan como una entidad única. Los programas utilizan cadenas para almacenar y transmitir datos y comandos.

Cálculo: Cómputo, cuenta o investigación que se hace de alguna cosa por medio de operaciones matemáticas.

Certificado: En el ámbito informático, consiste en una clave pública y el nombre de su propietario emitido por una Autoridad Certificadora.

Cifrar: Proceso de camuflar un mensaje o datos de forma que se oculte su contenido.

Clave: Valor utilizado junto con un algoritmo para cifrar o descifrar información.

Clave de Cifrado: Palabra o mecanismo que se utiliza para cifrar los datos o la información.

Codificar: Hacer o tomar un cuerpo o conjunto de instrucciones metódica y sistemáticamente para realizar alguna aplicación.

Colisión: En términos informáticos, dícese al choque de la información.

Comunicación: Tener correspondencia o paso unas con otras.

Contraseña: (Password) Cadena de caracteres única que hay que suministrar para obtener la autorización de entrada a un sistema o aplicación.

Criptoanalizar: Técnica usada para descifrar textos cifrados por criptógrafos.

Criptógrafos: Personas encargadas de cifrar y proteger información confidencial de posibles atacantes.

Criptograma: Análoga a texto cifrado. Texto ininteligible.

Criptología: Ciencia que se encarga del estudio y practica de los sistemas de cifrado destinados a ocultar el contenido de los mensajes enviados entre Emisor y Receptor.

CriptoSistema: Análoga a Sistema de Cifrado. Aplicaciones basadas en algoritmos o problemas matemáticos de difícil solución de aplicaciones específicas a los procedimientos involucrados en dicha aplicación.

D

Dato Binario: Información codificada en ceros (0) y unos (1).

Dato: Antecedente necesario para llegar al conocimiento de alguna información.

Descifrar: Proceso contrario al de cifrar, el destinatario intenta saber el contenido del mensaje cifrado.

Descriptar: Algún intruso intenta sabotear o descubrir el texto cifrado

E

E-mail: Abreviatura de Correo (mai) Electrónico (electronic).

Emisor: Persona que envía información a través de la red con un destino.

ENIGMA: Maquina que utilizaron los nazis para el envío de información o mensajes encriptados.

Entidad: Constituye la esencia de algo o alguien. Colectividad considerada como unidad: empresa, sociedad etc.

Entorno Seguro: Considerado como medio, espacio de trabajo o almacenamiento de la información de confiable seguridad.

Estrategia: Arte o facilidad para dirigir algún proyecto.

Exploit: Programas utilizados para aprovechar los fallos de los bugs y atacar al sistema computarizado.

F

Fragmentar: Reducir a trozos o restos de información.

Fuente: Análogo a Emisor. Orígen del envío de datos.

I

Información: Conjunto de datos que se obtiene para resolver algún problema.

Interceptar: Apoderarse de la información antes de que llegue al lugar o a la persona que se destina. Obstruir la vía de comunicación entre Emisor y Receptor.

Interlocutor: Dicese de cada una de las personas que toman parte en un dialogo o comunicación.

Intruso: Dicese a la entidad que trata de perturbar en la comunicación de dos Interlocutores. Persona que quiere acceder sin permiso alguno a un sistema computarizado.

Investigar: Hacer diligencias para descubrir algunas cosas

J

Jeroglíficos: Aplicase a la escritura en la que no se representan las palabras con signos fonéticos sino el significado de las palabras con figuras o símbolos. Conjunto de signos y figuras con que se expresa una frase que hay que adivinar.

L

Longitud de la Clave: Numero de símbolos o signos que forman la Clave de Cifrado.

M

Mainframes: Ordenador o computadora de gran capacidad, diseñado para realizar tareas computacionales muy intensas.

Malware: Programa creado de forma intencionada para provocar algún daño a un sistema informático.

Mecanismo: Estructura de un cuerpo natural o artificial y combinación de sus partes constituyentes. Modo de operar.

Medidas: Disposición o prevención de algún problema.

Medio: Lugar donde se desenvuelven un conjunto de actividades.

Mensaje: Comunicación que envía una persona a otra.

Método: Procedimiento para llevar a cabo un fin.

N

Nomeclatura: Conjunto de las voces técnicas y propias de una ciencia o facultad.

O

Orden: Regla o modo que se observa para hacer las cosas. Serie o sucesión de las cosas.

P

Permutación: Variar la disposición y orden en que estaban dos o mas cosas (letras, palabras, frases).

Pirata Cibernético: Dicese a la persona que se dedica al robo de información en la red.

Política de Seguridad: Conjunto de medidas que se debe de tomar en cuenta a la hora de diseñar un sistema informático.

Proceso: Conjunto de las fases sucesivas de un proyecto.

Protocolo: Conjunto de reglas que gobiernan el formato y significado de ciertas aplicaciones que son intercambiadas por la correspondientes entidades dentro de una comunicación.

R

Receptáculo: Cavidad en que se contiene o puede contenerse cualquier cosa (información).

Receptor: Persona que recibe información a través de la red (Destino).

Red: Es un grupo de equipos conectados entre si, de forma que pueden compartir recursos como: software's, impresoras, archivos etc.

Riesgo: Contingencia o proximidad de que se produzca un daño.

S

Sabotaje: Daño o deterioro que intencionadamente se causa en computadores, empresas, y software's pertenecientes a particulares o al Estado por motivos políticos, laborales o sociales.

Secreto: Lo que cuidadosamente se tiene reservado y oculto.

Servidor: Computadora conectada a una red que pone sus recursos a disposición del resto de los integrantes de la red. Suele utilizarse para mantener datos centralizados o para gestionar recursos compartidos

Símbolo: Figura con que se representa un concepto, por alguna semejanza que el entendimiento percibe entre ambos. Letra o conjunto de letras, adoptadas por convenio con que se representa un elemento.

Signo: Cualquiera de los caracteres que se emplean en la escritura. El que lleva al conocimiento de una cosa por la analogía o dependencia natural que tiene con ella.

Sistema Computarizado: Cualquier conjunto de dispositivos que colaboran en la realización de una tarea.

T

Táctica: Arte que enseña a poner en orden las cosas. Conjunto de reglas a las que se ajustan en su ejecución las operaciones para realizar de manera rápida y eficaz el desarrollo de una aplicación.

Técnica: Conjunto de procedimientos de que se sirve una ciencia o un arte. La técnica consta de: creación, invención de procedimientos, mejora de los ya existentes, búsqueda de recursos y organización de todos ellos al objeto de conseguir la realización del fin perseguido.

Técnica de Encriptación: Procedimientos utilizados para el camuflaje de mensajes con el objeto de que solo los autorizados puedan leerlos.

Tecnología: Término general que se aplica al proceso a través del cual los seres humanos diseñan herramientas y máquinas para incrementar su control y su comprensión del entorno material.

Telecomunicación: Sistema de comunicación a distancia por medio de: la red (Internet), teléfono u otro procedimiento análogo.

Transacción: Trato, convenio o negocio.

U

Usuario: Persona experta en computadoras, particularmente en la gestión de aplicaciones, más que en programación o en el mantenimiento de hardware.

V

Versión: Modo que tiene cada uno de referir un mismo suceso. Cada una de las formas que adopta el texto de una obra o la interpretación de un texto según quien sea su autor.

Virus Informático: Secuencia de código que se inserta en un fichero ejecutable de forma que cuando el archivo se ejecuta, el virus también lo hace, insertándose a sí mismo en otros programas.

W

WWW: Siglas de World Wide Web, mecanismo proveedor de información electrónica para usuarios conectados a Internet.

Web Site: Un lugar de la Web que mantiene en ejecución un programa llamado “servidor de páginas Web” que procesa las peticiones de información, típicamente solicitudes de páginas. Cada documento en uno de estos lugares tiene asignada una dirección única denominada URL.

BIBLIOGRAFÌA

Libros:

1. Galéndez Díaz Juan Carlos, "***Criptografía (Historia de la escritura cifrada)"***, 1ra edición, Editorial Complutense, S.A., 1995.

2. Lucena José Manuel, "***Criptografía y Seguridad en Computadores"***, 3ra edición, Versión 1.14, Marzo 2002.
   Página Web: http//:www.kriptopolis.com/libro.html

3. Schneider Bruce, "***Applied Cryptography"***, John Wiley & Son, 2da edition, 1996.

4. Tanenbaum Andrew, "***Redes de Computadoras"***, Prentice Hall, 1997.

Manuales:

1. Alarcos Alcázar Bernardo, "***Seguridad de la Información"***, Ingeniería Telemática, Departamento de Automática.
   Página Web: http//:www.autalcala.es/~alarcos.
   E-Mail: bernardo@autalcala.es

2. Ángel Ángel José de Jesús, "***Data Encryption Stadard".***

3. Ángel Ángel José de Jesús, "***El Sistema RSA"***, No. 2.

4. Ángel Ángel José de Jesús, "***Generación de Números Pseudoaleatorios usados en Sistemas Criptográficos"***, No. 1.

5. Ángel Ángel José de Jesús, "***Sistemas Criptográficos Asimétricos basados en Curvas Elípticas"***.
   E-Mail: jesus@seguridata.com.mx

6. Martínez Barberá Humberto, "***Criptografía"***, Universidad de Murcia.

7. Martinez Barbera Humberto, "***Seguridad en Redes de Ordenadores"***, Universidad de Murcia.

8. Ramió Jorge, "***Aplicaciones Criptográficas"***, UPM, 1999.
   E-Mail: jramio@eui.upm.es

Páginas Web:

1. http//:www.certicom.com (Criptografía con Curvas Elípticas).
2. http//:www.comterpone.com (Bruce Schneider)
3. http//:www.criptored.com
   :www.criptored.upm.com
   www.dat.etsit.upm.es/~mmonjas/cripto/00.html (Miguel Ángel Monjas).
4. http//:www.cryptome.org
5. http//:www.html.net
   www.html.net/seguridad.cripto
6. http//:www.htmlweb.com
7. http//:www.kryptopolis.com
   www.kryptopolis.com/tri002.html (Alfredo Zurdo Zarza).
8. http//:www.lawebdelprogramdor.com
9. http//:www.monografias.com
10. http//:www.rediris.es
11. http//:www.RSACrytography.com
12. http//:www.SistemasInformaticos.com
13. http//:www.toptutoriales.com
    www.toptutoriales.com/matematicas

Referencias en Línea:

En Inglés:

1. AES Home page.
2. Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip.
3. Cryptography Timeline Design (libro).
4. IDEA.
5. PGP 5.5, User's manual for Windows 95/NT.
6. RSA Data Security, Inc.
7. SSL 3.0 Specification.
8. Why Cryptography is harder than it looks.

En Español:

1. Boletín Criptonomicón.
2. Criptografía al servicio de las nuevas tecnologías.
3. Criptología y Seguridad.
4. Dinero Digital, por José Manuel Gómez.
5. Eligiendo el tamaño de la clave.
6. La seguridad de los protocolos criptográficos.
7. Los entresijos del firmado digital
8. Mecanismos de Seguridad.
9. Servidores Seguros.
10. Taller de Criptografía