UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

FACULTAD DE INGENIERÍA

DESARROLLO DE MECANISMOS DE SEGURIDAD DE INFORMACIÓN EN REDES DE DATOS PARA LA CREACIÓN DE UN SITIO DE COMERCIO ELECTRÓNICO.

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERÍA

INGENIERÍA ELÉCTRICA - TELECOMUNICACIONES

P R E S E N T A:

JOSÉ MARÍA RODRIGO CASTILLO ALAMILLA

DIRECTOR DE TESIS:
Dr. Francisco Javier García Ugalde

CIUDAD UNIVERSITARIA Octubre de 2006

JURADO ASIGNADO:

Presidente: Dr. Gómez Castellanos Javier

Secretario: Dr. Gutiérrez Castrejón Ramón

Vocal: Dr. García Ugalde Francisco

1er. Suplente: Dr. Rangel Licea Víctor

2do. Suplente: Dr. Rivera Rivera Carlos

Lugar o lugares donde se realizó la tesis:

División de Estudios de Posgrado de la Facultad de Ingeniería.

DIRECTOR DE TESIS:

Dr. García Ugalde Francisco

A la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO y a la Facultad de Ingeniería División de Estudios de Posgrado por darme la oportunidad de culminar otra meta en mi carrera profesional

A MI TUTOR Y DIRECTOR DE TESIS:

Dr. Francisco J. García Ugalde

Como un testimonio de gratitud por su inestimable ayuda para la culminación de este trabajo.

G R A C I A S

A MIS PADRES:

ELDA ALAMILLA MARTÍNEZ

RAFAEL CASTILLO VÁZQUEZ

Con todo cariño y admiración por estar conmigo siempre, brindándome su apoyo en todos los aspectos. Por su amor, comprensión y ejemplo.

A MIS HERMANOS:

HUGO Y DIEGO

Quienes son parte fundamental en mi vida, por su cariño y apoyo.

JOSÉ MARÍA

Índice

Objetivo:

Desarrollo de una metodología para el diseño de un sitio de comercio electrónico, la cual permite evidenciar los errores de seguridad más comunes que se cometen al generar aplicaciones de comercio electrónico, así como la mejor manera de evitarlos. Se analizan numerosas herramientas, técnicas y alternativas para que el desarrollador cuente con un amplio abanico de recursos y pueda utilizar posteriormente lo más adecuado en función de las características del problema concreto. Se hace especial incidencia en las ventajas respecto a la seguridad de un buen diseño de la aplicación en las tres capas: de presentación, lógica de negocio y datos. Con el propósito de concretizar la metodología propuesta en esta tesis, se propone un sitio de comercio electrónico en el que se ejemplifica la protección de la información a través de algoritmos de cifrado, la evolución de Internet y algunas metodologías comerciales actualmente en uso. Con todo esto, se estructura la seguridad tanto a nivel de lenguajes de programación, como a nivel de dispositivos.

RESUMEN

El presente documento de tesis intenta analizar uno de los aspectos más importantes en lo que se refiere a las aplicaciones de negocios sobre *Internet* como lo es la seguridad y confiabilidad en el comercio electrónico. Por tal motivo se examinan varios aspectos indispensables para un correcto funcionamiento como lo son: la seguridad en redes, la protección de la información a través de algoritmos de cifrado, las herramientas necesarias para una adecuada implementación, así como la de proporcionar un canal seguro para las transacciones entre los diversos entes que participan en el comercio electrónico.

Es un hecho que *Internet* constituye un canal de comunicaciones inseguro, debido a que la información que circula a través de esta extensa red es fácilmente accesible en cualquier punto intermedio por un posible atacante. Los datos transmitidos entre dos nodos de *Internet* se segmentan en pequeños paquetes que son encaminados a través de un número variable de nodos intermedios hasta que alcanzan su destino. En cualquiera de ellos es posible leer el contenido de los paquetes, destruirlo e incluso modificarlo, posibilitando todo tipo de ataques contra la confidencialidad y la integridad de los datos, y la actividad de comercio electrónico al depender de estos canales inseguros esta propensa a ser atacada por lo que se requieren de mecanismos de seguridad específicos, como vienen siendo algoritmos criptográficos ya sea de clave simétrica como de clave asimétrica y a partir de estos crear “certificados” que nos permitan poder realizar actividades comerciales mas seguras. Asimismo se emplea la criptografía para la protección de bienes informáticos vendidos a través de *Internet* y que estos no puedan ser “clonados” indiscriminadamente por los usuarios sin la autorización del autor. Aparte de utilizar mecanismos criptográficos también se profundiza en otros aspectos de seguridad como son la protección y acceso a la información; medidas necesarias para el alojamiento de un portal seguro, mediante programas o dispositivos específicos como son: *firewalls* o cortafuegos, sistemas detectores de intrusos, programas antivirus, etc.

Por otra parte, a manera de ejemplo de desarrollo, se creó un modelo de tienda virtual para el comercio electrónico. De tal modo que se emula un portal comercial en el que se muestra todo un análisis de riesgos y debilidades de seguridad informática posibles para este negocio. También se hace un estudio de los requerimientos de *software* y *hardware*, criptográfico y no criptográfico empleados en la elaboración de este sitio *Web*, haciendo énfasis en las ventajas que estos presentan, en la adecuación y utilización de este proyecto.

Posteriormente se prosiguió a desarrollar una serie de interfases y aplicaciones requeridas para el cifrado de información donde una página *Web* es utilizada para cifrar archivos usando algoritmos conocidos como DES, Blowfish o Rijndael que permiten resguardar la información y que esta solo sea desprotegida o reproducida por el programa adecuado; para la reproducción se hizo uso de las capacidades multimedia de la arquitectura Java para la creación de un programa que nos permita reproducir solo los archivos autorizados.

La presente tesis se basa en la seguridad del comercio electrónico, sin embargo la metodología aquí desarrollada puede servir para otro tipo de aplicaciones.

CAPÍTULO 1

INTRODUCCIÓN

Introducción

Las aplicaciones en *Internet* en general y de comercio electrónico en particular se están convirtiendo en un importante motor de la economía digital que se debe integrar al programa general de mercadotecnia de las empresas, para apoyar en la construcción de identidad de marca y aumentar las ventas, de la misma forma en que se usan las relaciones públicas, la publicidad, el correo directo y las llamadas telefónicas.

Internet ofrece un nuevo mercado que define la "economía digital". Los productores, proveedores de bienes/servicios y usuarios logran tener acceso y transmisión mundial de la información y esparcimiento en forma sencilla y económica, sean con fines comerciales o sociales. La apertura de mercados es fundamental para el rápido crecimiento del uso de nuevos servicios y la asimilación de tecnologías nuevas. En la práctica, las empresas están comenzando a usar *Internet* como un nuevo canal de ventas, sustituyendo las visitas personales, correo y teléfono por pedidos electrónicos, ya que gestionar un pedido por *Internet* cuesta menos que hacerlo por vías tradicionales. Surge entonces el comercio electrónico, como una alternativa de reducción de costos y una herramienta fundamental en el desempeño empresarial.

Sin embargo, la aparición del comercio electrónico obliga claramente a replantearse muchas de las cuestiones del comercio tradicional, surgiendo nuevos problemas, e incluso agudizando algunos de los ya existentes. En ese catálogo de problemas, se encuentran la seguridad; la fiabilidad del vendedor y del comprador en una relación electrónica, la falta de seguridad de las transacciones y medios de pago electrónicos, la falta de estándares consolidados, la proliferación de aplicaciones y protocolos de comercio electrónico incompatibles y la congestión de *Internet*.

Según numerosos estudios, la seguridad es el aspecto que más limita el crecimiento de la actividad económica en la Red. Los sistemas de información en línea y de realización de transacciones económicas a través de medios de telecomunicación se enfrentan a numerosas amenazas. Los ataques de piratas contra las tecnologías de la información son cada vez más frecuentes y sus efectos más devastadores. Por estos motivos resulta de crucial importancia poseer la capacidad de detectar y bloquear ataques en tiempo real contra los propios sistemas de información.

Estudios realizados durante los últimos años por diversas compañías dedicadas a la seguridad en *Internet*, indican que tan solo un 10 por ciento de las aplicaciones *Web* pueden considerarse seguras ante cualquier tipo de ataque [42]. En estos datos se incluyen sitios de comercio electrónico, banca online, B2B, sitios de la Administración, etc. Los estudios realizados han concluido que al menos un 92% de las aplicaciones *Web* son vulnerables a algún tipo de ataque [42].

El proceso de diseñar un sistema de seguridad podría decirse que es el encaminado a cerrar las posibles vías de ataque, lo cual hace imprescindible un profundo conocimiento acerca de las debilidades que los atacantes aprovechan, y del modo en que lo hacen. Además, existe una gran variedad de ataques posibles a vulnerabilidades incluso en la práctica se utiliza una combinación de éstas.

Los intrusos, antes de poder atacar una red de datos encaminada para el comercio electrónico, deben obtener la mayor información posible acerca de esta; intentan obtener el tipo de seguridad implementada, la topología, el rango de IPs de la red, los S.O, los nombres de usuarios, etc.

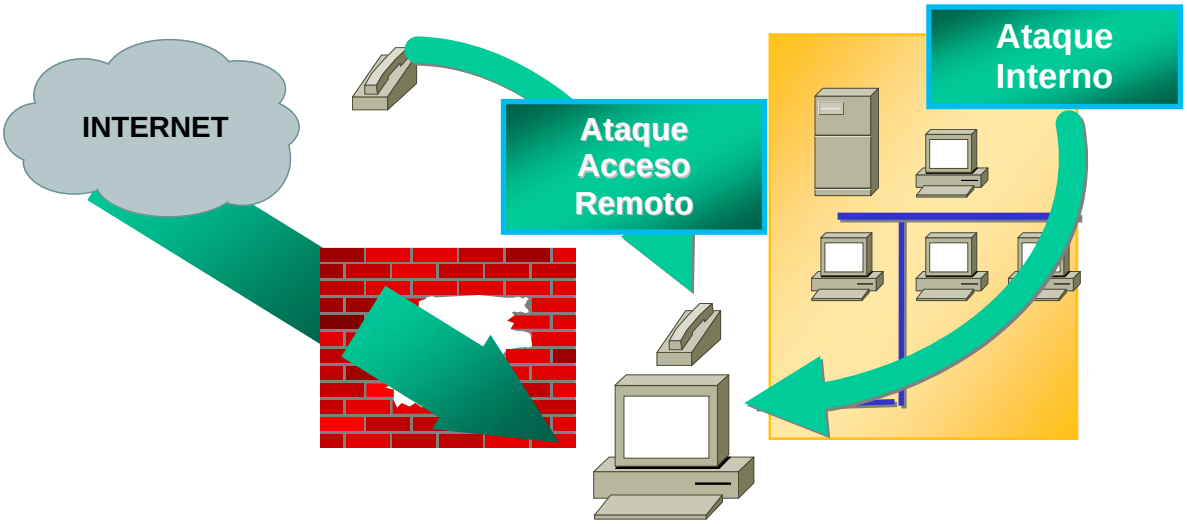


Figura 1-1. Ataque a una Red de Datos.

Contando con la formación adecuada, estos ataques podrían evitarse en la fase de desarrollo, sin necesidad de recurrir a costosas soluciones. Es por este motivo que la presente tesis va encaminada al desarrollo de mecanismos de seguridad involucrados en el desarrollo del comercio electrónico. Sin embargo el valor de este documento se basa en analizar y estructurar uno de los temas de mayor importancia en lo que se refiere a la seguridad de las aplicaciones de negocios sobre *Internet*.

La primera etapa de este trabajo consistió en examinar la situación actual del comercio electrónico así como hacer un análisis de los diversos mecanismos de seguridad necesarios para la elaboración de un sitio de comercio electrónico y las posibles vulnerabilidades a las que se enfrenta la seguridad informática existente en esta actividad.

La siguiente etapa consistió en estudiar las diversas tecnologías y lenguajes de programación necesarios para el desarrollo de todo un entorno de seguridad en la actividad comercial en *Internet* como son:

- La creación de un entorno confiable para el portal electrónico
- El desarrollo de las aplicaciones criptográficas para proteger el bien a vender

ya que dependiendo de las cualidades y estudio que estas tenían en cuestión de herramientas de seguridad, así como las ventajas que cada una ofrecía se planteo la estrategia a seguir para de este modo ofrecer un mejor desempeño en la elaboración de este trabajo.

Como tercera etapa se creó un modelo de tienda virtual para el comercio electrónico, a grandes rasgos se planteo un sistema que emulará un portal comercial cualquiera. Para tal efecto, se diseño un sitio comercial para venta de música por las características del mercado, economía y disponibilidad de ancho de banda para los cibernautas en México. A

partir de ahí, se desarrollo un modelo en el cual se sintetiza todo el análisis de riesgos y debilidades de seguridad informática posibles para este negocio; para luego proponer los requerimientos de "*software*" y "*hardware*", criptográfico y no criptográfico, necesarios para cubrir todas las necesidades detectadas.

La etapa final fue la creación de aplicaciones para la ejecución en sistemas locales, que permitieran ejemplificar el uso de herramientas criptográficas para proteger bienes informáticos vendidos a través del comercio electrónico y de este modo preservar el valor del bien y que este no sea susceptible a ser clonado para darle un mal uso.

1.1 La Nueva Economía

El impacto de nuevas tecnologías ha sido tan dramático que se ha creado de hecho una clara división entre los negocios tradicionales, de estructuras y organizaciones convencionales, de tiendas que existen físicamente y de relación directa con el cliente, y los negocios de la nueva era, las empresas de la nueva economía con robustos Sitios *Web*, negocios optimizados en su logística, servicio y garantías. El comercio electrónico, el dinero electrónico, los monederos electrónicos, y otras formas electrónicas de conceptos tradicionales, son términos que ya empiezan a ser reconocidos cotidianamente, y que poco a poco se irán intercalando en el uso y costumbres sociales y económicas.

En diversos sectores, las empresas que funcionan a través del comercio electrónico han venido a desplazar a los negocios convencionales al ofrecer un bien o servicio, de una forma conveniente para el cliente, a un precio accesible y con un servicio y atención que difícilmente puede ser superada por un negocio convencional. Y es que en el comercio electrónico en línea, el personal operador del sitio puede ofrecer al comprador, una atención mucho más personalizada, al conocer el nombre, dirección, teléfono, e-mail, preferencias, historial de compras, experiencia de pago, entre otros conceptos, de cada uno de sus clientes.

En algunas ciudades de alta concentración de población, como es el caso de las principales ciudades de México, la gente comienza a evitar desplazarse entre zonas que resulten lejanas. Por esta razón, la gente empieza a interesarse por las nuevas ofertas de bienes y servicios a través del *Web* para la adquisición de productos tales como ropa, artículos deportivos, enceres menores, perfumería y regalos, música, juguetes, computación y electrónica, reservaciones para hotel y transporte, e inclusive productos básicos no perecederos.

1.1.1 Aplicación del Comercio Electrónico en la Banca

Toda transacción que el usuario haga electrónica y remotamente, se considera banca electrónica o e-*banking*. Bajo este esquema se cobijan servicios ya tradicionales como EDI (*Electronic Data Interchange*, es el primer estándar ampliamente aceptado para la transferencia de transacciones de negocios), cajeros automáticos y puntos de ventas, o servicios novedosos como *callcenter*, *Homebanking* e *Internetbanking*, cuyos puntos de acceso (PC, *Laptop*, televisión interactiva, celulares *wap*, etc.) están en manos del cliente.

Figura 1-1. Ataque a una Red de Datos.

Cada banco esta formado por sus sucursales, su centro de información y sus redes de conexión. Las sucursales se conectan a su centro de información por medio de una *WAN* propietaria en la cual todas las transferencias se realizan de forma segura. Además su centro de información esta constituido por varias redes *LAN* y servidores *Web* y *EDI*. Todo esto en conjunto forma el medio interbancario.

Los clientes de cada banco pueden hacer operaciones vía *Web* o EDI, donde dichas actividades son en tiempo real si las cuentas involucradas pertenecen al mismo banco, ya que se realizan únicamente dentro del medio intrabancario.

1.2 Estructura del Documento

Este documento esta conformado por cinco capítulos, una sección de: conclusiones, anexos, un glosario y un apartado bibliográfico y de referencias. A continuación se describe brevemente el contenido de cada uno.

El primer capítulo es la introducción al presente documento: muestra la motivación del trabajo, la estructura del documento y la terminología usada en él; y el marco de referencia en el cual se desarrolla el comercio electrónico y la presente tesis. A grandes rasgos se dan los conceptos básicos de la información, como ha evolucionado hasta nuestros días, su asimilación en la vida cotidiana, y su relación con el comercio electrónico.

En el segundo capitulo se hace un análisis del comercio electrónico que va desde la situación que esta viviendo en la actualidad hasta la necesidad de implementación de seguridad para el desarrollo de esta actividad. Se detallan los diferentes tipos de comercio electrónico posibles, las etapas evolutivas de su implantación y los principales

componentes. De igual forma se muestra una metodología, para evaluar qué se debe proteger en el comercio electrónico y se estudian los modos de operación entre los diversos participantes que se involucran en esta actividad.

La seguridad de la información y la criptografía es el tema del tercer capítulo. Se muestran los conceptos básicos y objetivos de la seguridad de la información y la criptografía, al referir a la seguridad de la información no solo se analizo desde la simple protección de los datos a nivel lógico si no desde una perspectiva mucho mas extensa donde se tienen que tener en cuenta múltiples factores tanto internos como externos para poder identificar las amenazas, con respecto a la criptografía se hizo un estudio de algoritmos simétricos como asimétricos, tipos de ataques, sistemas de intercambio de llaves, y la noción y uso de los certificados digitales. Se detalla el uso algunos de algoritmos, tanto por su relevancia como por su uso en el presente trabajo.

El capítulo cuarto es una conjunción de los dos capítulos anteriores, donde se entrelazan el comercio electrónico con la criptografía y la seguridad de la información, es en este capítulo donde se describen los mecanismos de seguridad implementados en la creación del portal electrónico, cubriendo los aspectos más significativos al analizar los ataques y sus correspondientes defensas. También se hace un estudio de las tecnologías y lenguajes de programación implementados, describiendo su utilidad y modo de operación; justificando su uso en el escenario en el cual se va a desarrollar y con las circunstancias actuales de México.

En el capítulo cinco se presenta el desempeño de los ejemplos desarrollados así como los acontecimientos surgidos durante la fase de programación de las aplicaciones elaboradas. También se hace un análisis de los resultados más importantes obtenidos durante la realización de esta tesis; correspondientes a las etapas y actividades realizadas al momento integrar las herramientas criptográficas para resguardar la seguridad de la información utilizada en las transacciones de comercio electrónico.

También hay una sección de conclusiones donde se plantean las soluciones y vicisitudes que se obtuvieron en la realización de este trabajo, y como el comercio electrónico es una actividad que esta en continua evolución.

Además se tiene una sección de anexos que le permiten al lector conocer con mayor profundidad algunos de los temas importantes de esta tesis, sin la necesidad de recurrir a otras fuentes. Asimismo el presente cuenta con un pequeño glosario con los términos, y sus significados, más utilizados en el ciberespacio y en la seguridad informática.

Al final se tiene la bibliografía y referencias electrónicas utilizadas durante el desarrollo del trabajo de tesis. Esta sección puede ser de utilidad como guía de consulta para los lectores interesados en profundizar en un tema específico, debido a que se utilizaron fuentes de información disponibles al público.

1.3 La Seguridad en el Comercio Electrónico

El motivo de esta tesis es intentar analizar uno de los temas de mayor importancia en lo que se refiere a las aplicaciones de negocios sobre *Internet*: la búsqueda de mecanismos

que garanticen la seguridad y confiabilidad en el comercio electrónico. Esta nueva forma de hacer negocios, es un área de gran interés en una sociedad cuyas actividades y procesos cada día se basan más en la información electrónica, y en la cual se prevé a corto plazo: una creciente penetración del *Internet* comercial en la vida diaria y un incremento considerable en la utilización de sistemas de compras electrónicas.

Algunas de las claves para que el comercio electrónico llegue a ser seguro pueden encontrarse a continuación:

- Medios de pago adecuados. Una parte muy importante del potencial del comercio electrónico reside en la posibilidad de comercializar información especializada "a la carta". La naturaleza inmaterial del objeto de este tipo de transacciones, la gran cantidad de las mismas y la pequeña cuantía económica que representa cada una de ellas hacen que tanto los medios de pago tradicionales como su adaptación a *Internet* sean insatisfactorios para este cometido.

- Identificación y responsabilización de los usuarios. Es necesario proporcionar mecanismos de identificación de los usuarios (tanto clientes como proveedores) y manejo de la confianza entre los mismos. Está identificación puede o no ser análoga a la que se realiza en el mundo real. En muchos casos será suficiente con una identificación tipo autorización (cómo la que figura en un cheque al portador), mientras que otras veces se necesitarán medios de identificación más sofisticados.

- Mecanismos de protección de los elementos privados. Sean estos una imagen, un documento, información en un determinado directorio de un servidor, objetos de un sistema distribuido, cuentas de correo electrónico o estadísticas de acceso y uso, es necesario definir mecanismos fiables para controlar el acceso a estos recursos, evitar su uso indebido, proteger los derechos de autor, etc.

- Anonimato. Las soluciones aportadas deben respetar la privacidad o el anonimato cuando este sea lícito.

1.3.1 Mecanismos de Seguridad

En México, la seguridad tiende a ser una prioridad para empresas y personas, ya que los delitos informáticos se multiplican día con día, y ante esta situación la tecnología de la información se esta convirtiendo en la perfecta aliada de las dependencias de gobierno, instituciones financieras y demás empresas en general que efectúan alguna actividad en el comercio electrónico.

Para hacer frente a las amenazas a la seguridad del sistema se definen una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno o varios mecanismos de seguridad.

Las estructuras de seguridad de un sitio de comercio electrónico tienen que cubrir varios aspectos adicionales a las que se emplearía en un sitio tradicional, uno de ellos es la implementación de un canal seguro para la realización de transacciones.

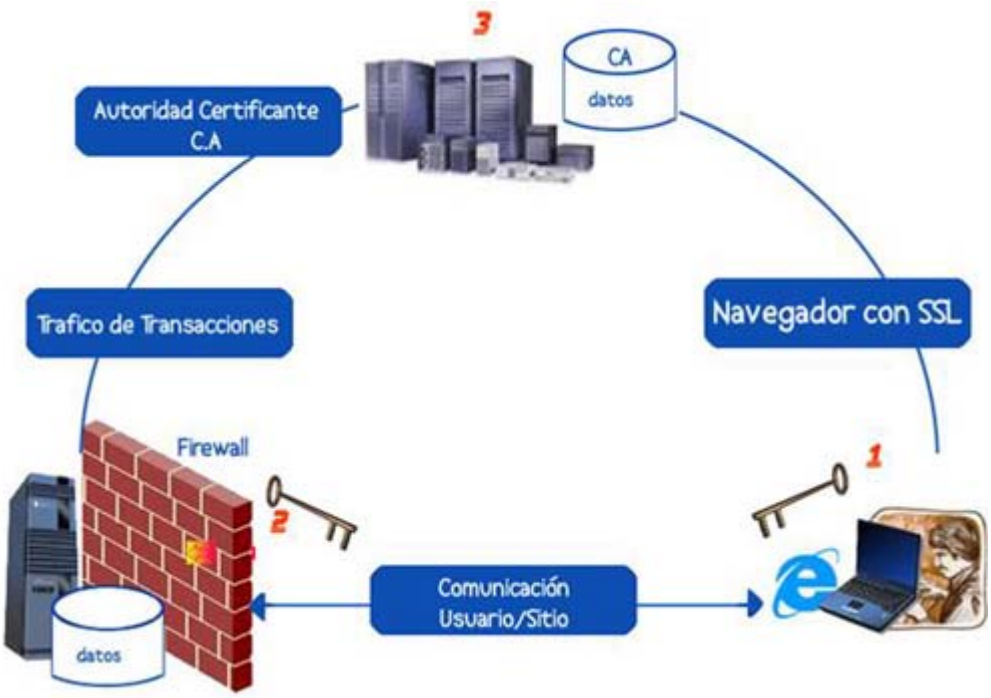


Figura 1-2. Seguridad de la Información en el Comercio Electrónico

1 Usuario conectándose a Punto2 utilizando un navegador *Internet* compatible con el protocolo SSL.
2 El Punto 2 es un sitio de comercio electrónico habitual (compra / venta) que establece conexiones seguras utilizando SSL para la transacciones, y también posee un *Firewall* para hacer filtrado de paquetes (Packet Filtering).
3 Este punto es la autoridad que emite los Certificados de Autenticidad (Certificate Authority) (CA), que por seguridad es recomendable que sea una tercera empresa el emisor del certificado no sea interno, sin embargo para el desarrollo de este trabajo y solo con fines de estudio en este documento se crea un certificado para la implementación del canal seguro.

El *firewall* es una herramienta preventiva contra ataques, que realiza una inspección del tráfico entrante y saliente. Esto impide que servicios o dispositivos no autorizados accedan a ciertos recursos y de esta manera protegernos contra ataques de denegación de servicios por ejemplo (DoS). El *Firewall* puede ser por Software o Hardware o bien combinaciones de estos.

Un certificado digital es un "pasaporte" electrónico el cual establece las credenciales de un sitio por medio del cual se realicen pagos, compras u otras transacciones a través de *Internet*. Un certificado es emitido por una autoridad certificada y contiene el nombre del sitio, un número de serie, fecha de expiración, una copia de la llave pública del dueño del certificado (utilizada para cifrar y descifrar tanto mensajes como firmas digitales), y la firma digital de la autoridad emisora del certificado por ejemplo *VeriSign* (empresa de seguridad informática famosa por ser una autoridad de certificación reconocida mundialmente. Emite certificados digitales para su uso en las transmisiones seguras por SSL (protocolo de transporte punto a punto de *Internet*), principalmente para la protección de sitios en

Internet en su acceso por https) de forma que el receptor pueda verificar que el certificado es real.

Este método de seguridad es óptimo ya que por cada conexión que se hace el servidor envía una clave diferente. Por lo tanto si alguien consigue desencriptar la clave lo único que podrá hacer es cerrar la conexión que corresponde a esa clave. También tiene grandes beneficios, ya que es un estándar no hace falta instalar ningún software adicional de lado del cliente, y tampoco de lado del servidor, ya que la mayoría de los servidores *Web* lo soportan para conexiones seguras. Los navegadores de *Internet* más populares ya poseen soporte para SSL. Debido a esto el 95% de los pagos de *Internet* se realizan utilizando hoy en día SSL.

Aparte de SSL existen otros protocolos, como el SET creado por *Mastercard* y *Visa* junto con líderes de la informática como *Microsoft*, *Verisign* y otras empresas más, que junto con el *CyberCash* son soluciones creadas para la venta por *Internet*, aunque en la actualidad todavía no tienen mucha difusión, por tal motivo en la presente tesis solo se profundizara sobre el protocolo SSL

Por lo anterior, esta sociedad en crecimiento, en la cual se almacena, procesa y transmite la información en forma electrónica, necesita las herramientas que le permitan salvaguardar esta forma de hacer negocio, buscando que dicha defensa resida en la propia información digital y sea lo menos dependiente del medio físico en el cual se encuentre. La seguridad informática y concretamente la criptografía son componentes críticos y de gran difusión para este modo de vida intensamente informatizado.

CAPÍTULO 2

MARCO DE REFERENCIA PARA EL COMERCIO ELECTRÓNICO

Marco de Referencia para el Comercio Electrónico

Desde su origen más primitivo, el comercio ha sido definido básicamente como un intercambio de bienes y servicios, evolucionando desde el llamado "trueque" hasta las formas contractuales más complejas existentes en la actualidad.

Antes de la aparición de las telecomunicaciones, las transacciones comerciales se realizaban siempre con limitaciones del espacio geográfico, requiriendo la cercanía física de las partes o de sus representantes. A medida que compradores y vendedores se alejaban, inventos como el teléfono y el fax, facilitaron el comercio haciéndolo más ágil y dinámico al permitir la gestión de los negocios a distancia en los casos en que se podía prescindir de la presencia física.

El avance en las comunicaciones a través de nuevas tecnologías como las redes computacionales, ha permitido una nueva forma de llevar a cabo transacciones comerciales a distancia, haciendo inválidas las limitaciones de espacio. Este fenómeno que comenzó con el Intercambio Electrónico de Datos (EDI) es más evidente en la *Internet*, ya que a través de sus servicios de World Wide Web y de correo electrónico permite a empresas y consumidores encontrarse en el ciberespacio y dentro de este espacio público virtual, llevar a cabo acciones de comercio. Es esta actividad que se conoce con el nombre de "comercio electrónico", involucra el conjunto de relaciones, transacciones y contratos comerciales que se dan parcial, o totalmente dentro del contexto de una red de telecomunicaciones, sea *Internet* o cualquier otra. La forma más avanzada es aquella que va desde la compra-venta, hasta la adquisición del producto en línea, pero también pueden darse etapas de la negociación por fuera de la red.

Con el comercio electrónico estamos frente a otra revolución del mundo de los negocios y ya se habla de una nueva economía que día a día adquiere más fuerza y en la que los principales participantes son los organismos multinacionales, los gobiernos nacionales, los sectores representativos, los proveedores de tecnología, las empresas y los consumidores, quienes se desplazan en ese gran centro comercial llamado *Internet*, considerada la puerta de entrada al futuro de la nueva economía global.

2.1 Definición de Comercio Electrónico

El comercio electrónico según La Real Academia Española se define como:

> ***"Negociación que se hace comprando, vendiendo o intercambiando géneros o mercancías"*** [18].

La Organización Mundial del Comercio en su página Web define al comercio electrónico como ***la producción, publicidad, venta y distribución de productos a través de las redes de telecomunicaciones*** [41].

Tan vaga como la anterior, existe una serie de definiciones, ambiguas algunas, pero poco precisas y poco concretas, por lo que una definición más cercana a la realidad de comercio electrónico sería:

El comercio electrónico es la acción de realizar de forma electrónica transacciones comerciales. Está basado en el tratamiento y transmisión electrónica de datos, incluyendo texto, imágenes y video. El comercio electrónico comprende actividades muy diversas, como comercio electrónico de bienes y servicios, suministro en línea de contenidos digitales, transferencia electrónica de fondos, compraventa electrónica de acciones, conocimientos de embarques electrónicos, subastas, diseños y proyectos conjuntos, prestación de servicios en línea, contratación pública, comercialización directa al consumidor y servicios postventa [17].

La forma en como opera el comercio electrónico permite al cliente acceder a mayor información del producto que se esta interesado en adquirir, comprar bienes que de otra forma no podría, así como utilizar medios de financiación novedosos y sobre todo seguros.

2.1.1 Situación actual

Las empresas se enfrentan hoy a competidores de todo el mundo. Todos estos fabricantes emplean la misma materia prima, la mecanizan con las mismas máquinas herramientas, tienen procesos de producción parecidos y soportan costos de transporte semejantes. Las diferencias entre ellas son y serán los procedimientos que hacen uso intenso de la informática y que resultan beneficiados con los procesos digitales [11].

Ahora, en este mercado global, la diferencia más importante entre las empresas consiste en como realizan su trabajo utilizando la información. Ganar o perder dependerá de cómo capten, gestionen y utilicen la información para saber si necesitan aumentar sus controles de calidad, rediseñar sus sistemas de producción, mejorar su publicidad, modificar la interacción entre distintos departamentos, cambiar sus canales de venta, o transformar su asistencia técnica, etcétera. Los ganadores serán los que integren todos sus sistemas y herramientas informáticas en un solo gran sistema, de manera que la información circule con facilidad dentro y fuera de sus empresas y se maximice constantemente el conocimiento [11].

Las empresas triunfadoras de este inicio de siglo serán las que utilicen los medios digitales para reinventar su propio funcionamiento. Esas compañías actuarán con eficiencia y hallarán vías positivas de contacto directo con sus clientes y con su entorno para detectar los cambios [11].

2.1.2 Aceptación del comercio electrónico

Los consumidores han aceptado paulatinamente el negocio de comercio electrónico, quizás más lentamente de lo que esperaban sus promotores. Incluso en categorías de producto aptas para el comercio electrónico, la compra electrónica se ha desarrollado lentamente. Muchas razones se pueden argumentar para esta lenta implantación, como son:

- Preocupación sobre la seguridad. Mucha gente no utilizará las tarjetas de crédito en *internet* debido a su preocupación sobre un posible robo, o fraude.

- Falta de gratificación instantánea en la compra (compras no digitales). Mucha recompensa obtenida por el consumidor en la compra reside en la gratificación instantánea que supone la utilización del producto. Esta recompensa no existe cuando la compra tarda en llegar días, o semanas.
- El problema del acceso a la Web, particularmente para hogares pobres, o países subdesarrollados. Las tasas bajas de penetración de *internet* en algunos sectores reducen el potencial del comercio electrónico.
- Aspecto social de la compra. Algunas personas les gusta hablar sobre el género con los dependientes, o acompañantes: esta recompensa social de la terapia comercial no existe en la misma dimensión en las compras online.

Sin embargo la influencia del comercio electrónico está notándose cada vez más tanto en las empresas beneficiadas por una presencia global y accesos a nuevos mercados y clientes; como en la sociedad, trayendo consigo una verdadera revolución en el ámbito del comercio global y la economía.

Los mismos medios electrónicos también han permitido el uso de la transacción híbrida, aquella en donde el comprador usa *Internet* para reunir información, pero realiza su compra por un canal diferente (pedido por teléfono, o visitando la tienda directamente) [7]. Estas transacciones no cuentan como comercio electrónico, aun cuando deben considerarse facilitadas por la Red.

2.1.3 Proceso de una transacción electrónica

Una transacción electrónica debe seguir las siguientes etapas [36]:

1. Ubicación de la oferta en línea: consistente en la colocación de la oferta dentro de una página Web utilizada por el vendedor, para dar a conocer sus diferentes productos. Puede recurrir a diversas herramientas disponibles, como la ubicación de su sitio en motores de búsqueda.
2. Proceso de selección de un pedido: a través de los diferentes catálogos o presentaciones que ubique al vendedor en su sitio Web, éste guiará al comprador para que logre tomar su decisión y efectuar el pedido.
3. Confirmación del pedido: una vez que el comprador se decide sobre los bienes y/o servicios que desea adquirir, el sitio Web deberá proveerlo de una opción que le permita visualizar cuál va a ser su pedido y las condiciones del mismo. Muchos desarrolladores de páginas Web utilizan un “carro de compras”, como una clara analogía de lo que el pedido va a ser en concreto.
4. Recepción de datos del pago: se deben sugerir las formas de pago electrónico disponibles, recordando en todo momento que a partir de que se acepte la oferta se configura el pago.
5. Procesamiento del pago: la página Web del vendedor lleva de la mano al comprador para que este de forma electrónica autorice a un procesador de tarjetas de crédito (o débito) a cargar en su cuenta el monto correspondiente al pedido seleccionado y perfeccionadote esa manera se concretiza el contrato de compra venta electrónica.
6. Confirmación del pago: simple formalidad para indicarle al comprador el monto debitado de su tarjeta de crédito (o débito), el número de transacción y el número de autorización (en los casos en que proceda), así como a nombre de quien

deberá aparecer el débito: en el estado de cuenta del comprador, o del tarjeta habiente que pagó la transacción y que concretizo el contrato.

7. Emisión de la factura o comprobante de adquisición: algunas jurisdicciones ya obligan a sus vendedores a la emisión de facturas o comprobantes que den fe de la existencia del contrato de compra venta electrónica entre las partes contratantes.

8. Envío o descarga de los bienes y/o servicios pactados: en el caso de las compras de bienes y/o servicios que deben ser entregados de forma electrónica, este paso permite la adquisición de los mismos y deberá contemplar la posibilidad para que el comprador pueda acceder a los archivos adquiridos, o en su defecto, pueda proceder a descargarlos en su computador.

Estas transacciones se realizan con la participación de 3 entes: negocios (organizaciones empresariales), consumidores (particulares) y la Administración (gobierno).

2.1.4 Beneficios del comercio electrónico

El negocio que vende sus productos a través de *Internet* se beneficia de una serie de ventajas que los canales tradicionales no ofrecen. Entre ellas se encuentran:

- Los costos de iniciar un sitio en el *Internet* son menores que los de instalar un establecimiento físico como punto de venta.
- Con el *Internet* se puede servir a los clientes de manera personal, permitiendo formar una relación estrecha con ellos y ayudar en el establecimiento de una estrategia de crecimiento futuro.
- Acceso al mercado global.
- Presencia mundial, a toda hora y todos los días.
- Menores costos de intermediación.
- Bajo costo en conseguir nuevos clientes a diferencia de medios de publicidad tradicionales (como la radio y TV).

2.1.5 Factores clave para el éxito en el desarrollo del comercio electrónico

Varios factores han tenido un importante papel en el éxito de las empresas de comercio electrónico. Entre ellos se encuentran:

- Proporcionar valor al cliente. Los vendedores pueden conseguirlo ofreciendo un producto o una línea de producto que atraiga clientes potenciales a un precio competitivo al igual que suceden en un mercado tradicional.
- Proporcionar servicio y ejecución. Ofrecimiento de una experiencia de compra amigable, interactiva tal como se podría alcanzar en una situación caracol presencia física.
- Proporcionar una página Web atractiva. El uso de colores, gráficos, animación, fotografías, tipografías y espacio en blanco puede aumentar el éxito en este sentido.
- Proporcionar un incentivo para los consumidores para comprar y volver. Las promociones de ventas pueden incluir cupones, ofertas especiales y descuentos.

Las paginas Web unidas por enlaces y los programas de publicidad pueden ayudar en este aspecto.

- Proporcionar atención personal. Paginas Web personalizadas, sugerencias de compra y ofertas especiales personalizadas pueden allanar el camino de sustituir el contacto personal que se puede encontrar en un punto de venta tradicional.
- Proporcionar un sentido de comunidad. Las áreas de *Chat*, foros, registro como cliente, esquemas de fidelidad y programas de afinidad pueden ayudar.
- Proporcionar confianza y seguridad. Servidores paralelos, redundancia de *hardware*, tecnología de seguridad en averías, encriptamiento de la información y cortafuegos pueden ampliar estos requisitos.
- Poseer la experiencia total del consumidor. Esto se consigue tratando con el consumidor como parte de una gran experiencia, lo que se hace ver como sinónimo de la marca.
- Optimizando los procesos de negocio, posiblemente a través de tecnologías de reingeniería de la información.
- Dejando que los consumidores se ayuden a sí mismos. Proporcionando sistemas de autoayuda.
- Ayudar a los consumidores a hacer el trabajo de consumir. Los vendedores pueden proporcionar esta ayuda ampliando la información comparativa y las búsquedas de producto. La provisión de información de componentes y comentarios de seguridad e higiene puede ayudar a los minoristas a definir el trabajo del comprador.
- Operar en, o cerca del límite de la tecnología y permanecer allí mientras la tecnología siga cambiando (pero recordando que los principios fundamentales del comercio se mantienen indiferentes a la tecnología)
- Construir una organización con suficiente agilidad y sistemas de alerta para responder rápidamente a los cambios en el entorno económico, social y físico.

2.2 Tipos de Comercio Electrónico

Existen diferentes criterios para clasificar al comercio electrónico:

1. Por el tipo de producto comercializado, entendiéndose por producto un bien, o un servicio.
2. Por el tipo de participantes en la transacción, en especial el tipo de negocio que está presente:

    - vende sus propios productos o servicios,
    - es un sitio de información,
    - vende productos o servicios de otras compañías, y
    - utiliza su sitio como otro canal de venta para complementar su negocio físico.

2.2.1 Tipos de comercio electrónico según el producto comercializado

Usando las definiciones de [10] se tiene la siguiente clasificación:

Comercio electrónico indirecto. Implica la manipulación de la información que se necesita para el comercio de bienes físicos. Maneja la publicidad, la investigación, la venta, la contratación y otras funciones relacionadas con la información, aunque los bienes reales sean objetos físicos remitidos según los sistemas tradicionales de transporte.

Comercio electrónico de bienes y/o servicios entregados de forma electrónica
Dentro de esta categoría se incluyen aquellos bienes y/o servicios que puedan ser digitalizados, manipulados, transferidos, enviados y almacenados vía electro-magnética, utilizando medios de telecomunicaciones.

Otra característica que presentan es la inmediatez del pago de las transacciones originadas para dar lugar al acto de comercio electrónico; su aceptación se dio en el mismo momento en que fue pagada y aceptada la oferta por parte del comprador, caso contrario, no hubiera podido realizar la descarga del archivo aludido y por lo tanto, no se hubiera completado el acto de comercio electrónico.

A continuación se definirá una serie de bienes y/o servicios que se pueden brindar bajo esta modalidad de contratación electrónica:

a) Contratos de venta de *software*: consisten en la venta de archivos ejecutables para descargar *software* disponible en la página Web del vendedor.
b) Contratos de licencia de uso de *software*: consisten en la venta de licencias para el uso de *software* comprado tangiblemente, o descargado de la página Web del vendedor.
c) Contratos de leasing sobre el *software*: consisten en el alquiler de archivos ejecutables para utilizar *software* disponible en la página Web del vendedor.
d) Contratos de licencia de uso de bases de datos: consisten en el acceso a las bases de datos disponibles en línea para los compradores, o en la venta de las mismas vía descargas de los archivos de la página Web del vendedor.
e) Contratos de almacenaje de bases de datos: consisten en el almacenamiento digital de diferente información en los servidores del vendedor.

La compra de un archivo digital que no involucre el pago directamente al momento de aceptar la oferta, no deberá ser considerada como un acto de comercio electrónico, sino como un acto de comercio de bienes electrónicos negociados de manera convencional.

Comercio electrónico de bienes y/o servicios entregados por medios no electrónicos
Bajo esta modalidad, se incluyen todas las transacciones negociadas en una página Web que no impliquen la transferencia de la propiedad de los bienes y/o servicios de forma electrónica, es decir, la negociación de bienes tangibles, siempre y cuando dicha negociación se sujete a los parámetros de oferta y aceptación electrónica, incluyendo dentro de esta última, el pago.

Es conveniente hacer notar, que aunque el pago de estas transacciones se haga en línea, no implica que la transacción no genere una obligación monetaria adicional, tal es el caso

de los aranceles aduaneros, cuando las transacciones se ejecuten entre jurisdicciones distintas.

Dentro de las transacciones electrónicas de bienes y/o servicios entregados por medios no electrónicos, se pueden citar los siguientes:

a) Venta de discos compactos de música y películas.
b) Venta de libros, revistas y periódicos.
c) Venta de computadoras y accesorios afines.
d) Venta de boletos aéreos.
e) Venta de comida enlatada.
f) Venta de bebidas.
g) Venta de servicios de traducción de documentos.
h) Venta de servicios de correduría aduanal.
i) Venta de servicios de pagos de impuestos.
j) Venta de servicios de correduría bursátil.
k) Venta de servicios de pago de servicios públicos.
l) Venta de servicios publicitarios.

La lista podría ser infinita, dependiendo de la evolución y desarrollo de las diferentes herramientas informáticas que van permitiendo día con día la inclusión y automatización de una mayor cantidad de procesos de venta de bienes y servicios.

2.2.2 Tipos de comercio electrónico según los participantes

Negocio a Consumidor ("*Business to Customer*", B2C). Las transacciones de empresa a consumidor son las que reciben más publicidad. Es el escenario clásico del comercio electrónico: navegando por la red, un consumidor puede elegir cómodamente desde su casa u oficina y a cualquier hora del día entre una infinita variedad de bienes y servicios de compañías distribuidas por todo el mundo. *Internet* facilita el acceso a productos a su alcance en el mercado local, y a otros que no existen en su mercado (productos informáticos, música, etcétera). En el caso de productos clasificados como digitalizados; pueden "descargarlos" de la red después de efectuar el pago con su tarjeta, disponiendo de ellos de forma inmediata.

El gran atractivo de este esquema para los compradores es la disminución de la cadena de intermediarios (distribuidores y minoristas) presentes en los esquemas tradicionales de comercialización. Al no existir tantos intermediarios, ya no se añaden los servicios y cargos correspondientes que se reflejaban en un aumento del precio final del bien o servicio adquirido por el cliente.

El gran atractivo para los negocios es la factibilidad de la relación fabricante-cliente final, tanto para productos como para servicios, ya que *Internet* le permite al primero:

- tener el equivalente a una puerta de venta directa de fábrica, aprovechando que la mayoría de las operaciones serán transacciones digitales en régimen de autoservicio [11]
- aumentar sus ventas al beneficiarse de las reducciones, debidas a la disminución de intermediarios

- explotar la enorme capacidad de este medio para difundir información valiosa a escaso costo y sin necesidad de abrir sucursales [11].

Pero la gran desventaja de este esquema es la espontaneidad de la relación vendedor-cliente: en cada transacción es más probable que los consumidores busquen el mejor trato (menor costo, entrega más rápida, etc.) sin importar con cual vendedor hayan comercializado en ocasiones anteriores.
Negocio a Negocio ("*Business to Business*", B2B). Este canal permite que las compañías vendan y paguen productos y servicios entre sí. Estructuralmente este canal es más atractivo que el B2C. Las barreras de ingreso son moderadas, los costos de cambio de proveedor son altos y la intensidad de la competencia entre titulares y nuevos participantes es modesta [7].

En un primer tiempo, esto ha marcado la gran diferencia entre B2C y el B2B: es más rentable para las compañías de *Internet* atender a organizaciones que a particulares. Los consumidores usan *Internet* para recopilar información que les ayude a lograr mejores tratos, presionando así las utilidades de los proveedores. Las compañías son clientes más rentables porque avanzan más lentamente que los consumidores. Las compañías prefieren crear un proceso de compras permanente una sola vez que reconsiderar y renegociar cada vez que hacen un nuevo pedido del producto, o servicio. Por tanto, las compañías tienden a tardar mucho más en tomar una decisión inicial de compra de un nuevo tipo de producto; enseguida, intentan estandarizar sus propios procesos con base en el producto, o servicio, del proveedor ganador. Incluso cuando aparece una tecnología nueva, se muestran renuentes a cambiar de proveedor [7].

Consumidor a Consumidor ("*Customer to Customer*", C2C). Son aquellos negocios que actúan como agentes intermediarios, o puntos de encuentro entre distintos particulares con el fin de ponerlos en contacto para el comercio entre ellos; obteniendo una comisión de las transacciones realizadas, y de la publicidad que permiten tener en su sitio de *Internet* [7].

Este esquema permite la reingeniería del ente intermediario, de una u otra forma desplazado por el B2C, al convertirlo en un suministrador de valor agregado mediante la distribución de información de los productos [11].

La tarea clave de los agentes intermediarios es poner en correspondencia las necesidades de cada consumidor-comprador con los bienes y servicios disponibles, y ofertados por los otros consumidores-vendedores. Para esta actividad se vuelven críticos los servicios de listas informatizadas que se caracterizarán por [10]:

- Tener bases de datos actualizadas con el estado del producto (en venta, liquidado, etc.) y de gran confiabilidad para manejar la complejidad que implica un enorme número de usuarios distintos.
- Contar con la capacidad de anexar imágenes, o datos multimedia a los bienes ofertados.
- Permitir tener toda la información asociada a los bienes (como ubicación, costo de mantenimiento, etc.).

Negocio a Gobierno ("*Business to Administration*", B2A). Los gobiernos del mundo son una categoría importante de las organizaciones humanas, con necesidades especiales y poderes especiales. Su actividad opera en dos niveles: dentro de las naciones y entre

ellas. Todo gobierno es gran comprador de bienes y servicios y, en consecuencia, gran candidato al comercio electrónico [10].

En este esquema prevalecerán las licitaciones electrónicas propuestas por el gobierno. Por esta razón se deberán estandarizar las formas electrónicas ("*e-forms*") con el fin de crear y usar herramientas informáticas que permitan al sector público ejecutar en forma rápida y ágil la comparación de distintas ofertas económicas en los distintos apartados y garantizar la equidad en la designación de los ganadores.

Además, si se analiza al gobierno como un cliente y proveedor de servicios, con características muy propias, se puede concluir que se ha desarrollado un mercado de la información gubernamental. En él se han hecho propuestas al gobierno para cubrir pedidos, facturas y procedimientos de conciliación, mantener programas y revisiones, etc. Estos procesos ya han empezado a reducir el costo de las compras gubernamentales y se prevé que esta tendencia continúe.

Ciudadano a Gobierno (*Citizen to Administration*, C2A). Por último los gobiernos necesitan comunicarse con sus ciudadanos. Su vida se basa en el empleo de formularios y de información estructurada, y siempre pueden permitirse ser más eficientes; razones por las cuales son excelentes candidatos a beneficiarse del mercado de la información. Los gobiernos pueden usar este enfoque para [10]:

- recibir propuestas de la gente,
- realizar la integración de los diferentes entes gubernamentales,
- realizar consultas y votaciones electrónicas; y
- manejar directamente el pago de impuestos de los ciudadanos.

Sobre todo esta última actividad se ha conceptualizado para:

1. La utilización de formas fiscales electrónicas ("*e-forms*" fiscales)
2. La realización de transacciones financieras electrónicas, de las cuentas de los individuos a las cuentas del gobierno.

2.3 Etapas Evolutivas en la Implantación del Comercio Electrónico

Las empresas, grandes y pequeñas, tienden a desarrollar su presencia en la Web en etapas, o fases. Una vez que una empresa ha hecho acto de presencia en *Internet*, entonces querrá usar ese sitio para reforzar el servicio al cliente y producir dividendos. En está fase es cuando el comercio electrónico entrará en acción. Muchos negocios pequeños y medianos están esforzándose con el alto costo de entrada al comercio electrónico. Creando un ambiente completo de ventas en línea que puede requerir tiempo, dinero y especialización técnica. Muchos negocios están detenidos en el primero, o segundo, de los tres niveles que llevan a construir una presencia efectiva en *Internet*, para un comercio electrónico eficaz.

Por ello, el desafío más importante es comprender los costos y beneficios progresivos de trabajar en *Internet*. La pirámide de evolución de las operaciones por la *Web* muestra las diversas etapas de la puesta en operación del comercio electrónico, a partir de cómo se realiza el flujo de información [7].

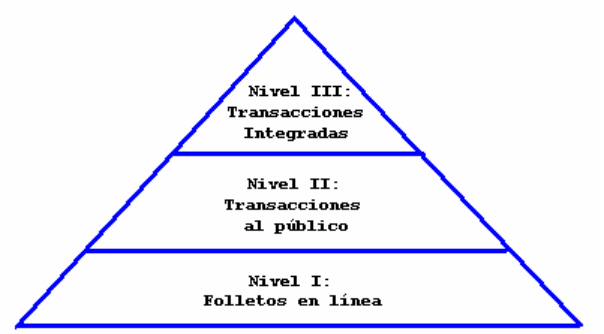


Figura 2-1. Pirámide de la Evolución de las Transacciones Comerciales por Internet

2.3.1 Nivel I: Etapa de folletos en línea

Se colocan en el sitio Web publicaciones sobre los productos, informes anuales y otro tipo de información tradicionalmente impresa. El flujo de información es unidireccional (compañía a clientes), pues se sigue utilizando un proceso concebido en papel. Como sólo tiene contenido, únicamente permite las transacciones híbridas, o fuera de línea [7].

- Ventajas: permite desarrollar los sitios "Web" fácil y rápidamente a bajo costo.
- Desventajas: esto limita la función de *Internet* a la promoción y ninguna oportunidad del crédito está envuelta.

2.3.2 Nivel II: Etapa de transacciones al público

Se utiliza el sitio "Web" como un medio de recepción de formularios de pedidos ("*e-forms*" propietarios, o normalizados). La información de pedidos recopilada por medio de *Internet* se imprime y emplea como entrada a un proceso inalterado de surtido de pedidos. Las compañías en este nivel no integran la información de los pedidos electrónicos directamente a sus procesos internos. El flujo de información es bidireccional (empresa -> cliente -> empresa) pero no es totalmente electrónico, ya que se imprime para seguir un proceso tradicionalmente concebido en papel.

- Ventajas: Permite las transacciones electrónicas únicamente para la compra-venta de productos. No necesita manejar tecnología sofisticada y su catálogo puede manejar un surtido de productos grande. Además permite el flujo de retroalimentación del cliente.
- Desventajas: La construcción del sitio Web (catálogo, herramientas de protección de transacción, etc.) incrementa el costo y puede que algunos de los servicios de valor agregado (atención en tiempo real, asistencia técnica, etc.) sólo se puedan manejar por otros canales.

2.3.3 Nivel III: Etapa de aplicaciones integradas

Hay un gran número de compañías que han instalado aplicaciones de transacciones integradas que explotan todo el poder de *Internet*. Estas aplicaciones usan la red para intercambiar información con los clientes. Dicha información está estrechamente relacionada con las operaciones internas de las compañías. El flujo de información es bidireccional, constante y electrónico, porque siempre se mantiene dentro de procesos digitalizados.

- Ventajas: permite manejar un surtido de productos grande y las ventas completas al más bajo costo; todas las transacciones son electrónicas y se permite usar este mismo canal para dar los servicios de valor agregado a todos los clientes.
- Desventajas: La construcción de un sistema de este tipo es la más costosa, y requiere de mayor administración ya que todas las transacciones y servicios de valor agregado requieren de tecnología sofisticada.

Como se observa, a medida que una empresa evolucione a través de estas etapas, la mayor parte de las interacciones con los consumidores no serán ventas, sino servicios y asistencia técnica [11].

2.4 Componentes de los Sistemas de Comercio Electrónico

Un sistema básico de comercio electrónico requiere que el cliente cuente con acceso a *Internet*, y que el vendedor cuente con el "*software*" para comercio electrónico (con el cual se crearán los catálogos de ventas y se procesarán las transacciones financieras); así como contar con un servidor de aplicaciones para la red, el cual debe contar con entradas de seguridad para limitar el acceso externo a los sistemas de datos, y "*software*" especializado que se encargará de "*lanzar*" los datos de los sistemas de apoyo apropiados en el ambiente del comercio (vea figura anexa).

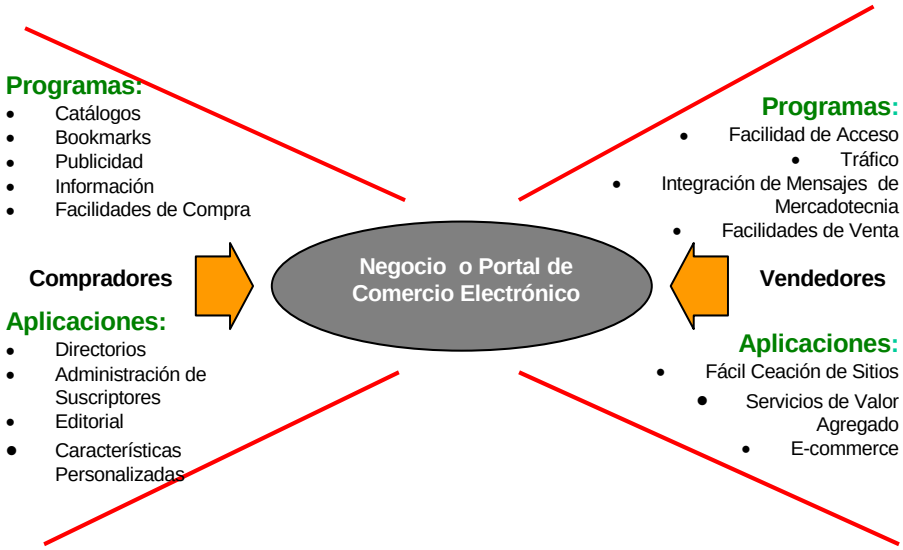


Figura 2-2. Componentes de los sistemas de Comercio Electrónico

2.4.1 Formulario electrónico

Una herramienta simple, pero poderosa es el formulario electrónico, o "*e-form*". Ésta permite estandarizar la información de las órdenes de compra y requisiciones de servicios; con la finalidad de automatizar en mayor o menor grado, la exploración, la negociación, los pedidos, la contratación y la facturación; y también permite reducir las barreras lingüísticas en el comercio internacional [10].

2.4.2 El servidor de transacciones

Un servidor de transacciones se ocupa de las operaciones de crédito y débito (usando tecnología estándar de seguridad electrónica) en nombre del comerciante y del cliente. Este servidor debe contener una Interfaz Programada para la Aplicación (API, *Application Program Interface*) que efectúe todos los tipos del pago y funciones como: recibir, aprobar, depositar y reintegrar.

El servidor de transacciones maneja la autorización necesaria; solicita y guarda la información de la transacción y liquida las transacciones del comerciante, la compañía de tarjetas de crédito, y el cliente. El servidor de transacciones maneja los procesos de pago y comunica la orden de pago generada por el consumidor a la institución financiera elegida por el comerciante. Deben mantenerse archivos de transacciones para facilitar la conciliación e información posterior.

El servidor de transacciones también debe contener un componente para procesar los certificados digitales de una organización que usa el "*software*" de autoridades certificadoras para permitir la evolución hacia tecnologías de seguridad mejoradas. Múltiples comerciantes podrían operar en un solo servidor de transacciones.

2.4.3 Los sistemas de pago

Los sistemas de pago requieren de componentes localizados en la ubicación del cliente final (casa, computadora personal, etc.), así como en el sitio en el que se ubica el sistema de transacciones del comerciante, y en el lugar donde reside la institución financiera.

Los consumidores deben estar seguros que su información financiera es confidencial; esto se cumple con carteras electrónicas o "*software*" de tarjeta de crédito en el punto extremo del consumidor. La información del crédito del consumidor se envía a un servidor de transacción que puede aceptar una variedad de pagos electrónicos, así como una tienda física puede aceptar el crédito, o información, de la tarjeta de débito.

Figura 2-3. La arquitectura distribuida del comercio electrónico

El servidor de transacción también debe manejar el proceso del pago, y se deberá comunicar con la institución financiera para liquidar los bienes adquiridos por el consumidor.

El servidor de transacciones mantiene la información de pago de transacción detallada, mientras permite a las compañías ocuparse de aclaraciones, rebotes de créditos, o ajustes.

2.4.4 Modelos de "hospedaje" para los diferentes tipos de sitio de comercio electrónico

Al implementar un sitio de comercio electrónico se tiene que tomar en cuenta la etapa o fase evolutiva en la que se encuentra la empresa en la red, ya que de acuerdo a esto se podrá decidir que modelo de "hospedaje" implementar en la tienda virtual de acuerdo a lo que requiera, ya sea instalando su propio servidor dedicado a sus productos, o contratando un proveedor de servicios de *Internet* (ISP's), que configure sus ofertas en cualquier combinación de los modelos siguientes, de tal manera que sus clientes (las empresas), se organicen en una plataforma tecnológica:

- *Hospedaje Simple*: El cliente es dueño de un sitio *Web* almacenado en un servidor de *Internet* compartido por muchos usuarios, el cual tiene un único URL. El cliente no realiza ninguna transacción en línea, pero tiene capacidad para correo electrónico y distribución de información y publicidad.
- *Hospedaje con Capacidad de Almacenamiento Adicional*: El cliente es dueño de una sola tienda en un solo servidor mercantil; es decir, toda su información se hospeda en un solo servidor. La tienda virtual posee un único URL, un banco de datos, y un proceso para registrar los cobros.
- *Centro comercial*: Se ofrece al cliente (o el cliente lo contrata para) tener múltiples tiendas en un ambiente tal que se asemeje a un centro comercial en el mismo

URL, presenta la ventaja adicional de poseer un solo banco de datos, con registros compartidos, un solo carrito, caja, etc.,

- *Multihospedaje*: Múltiples tiendas virtuales residen en un servidor, pero cada una de ellas tiene su propio URL, banco de datos, sistema de compra y formato de orden, etc.
- *Servidor de transacciones y contenido dentro del mismo sitio*: No hay una base de datos de transacciones, el sitio *Web* esta hospedado en varios servidores ("*multihome*") y las transacciones son llevadas a cabo por un servidor mercantil. Estos sitios *Web* de venta se crean con un botón de compra que envía al servidor por separado, la información del producto, y de la transacción (pero dentro de un mismo ambiente).

Las siguientes figuras ilustran los varios modelos de "hospedar" una tienda virtual.

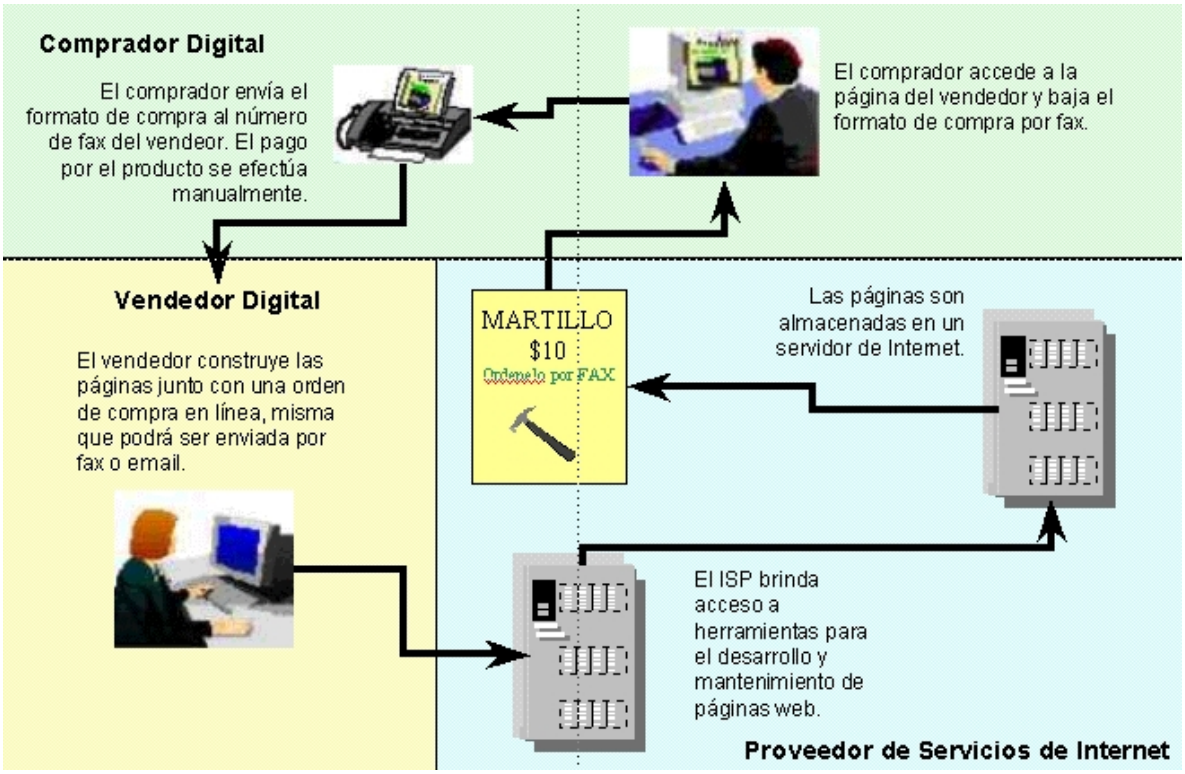


Figura 2-4. Modelo Simple de Hospedaje de Tienda Virtual

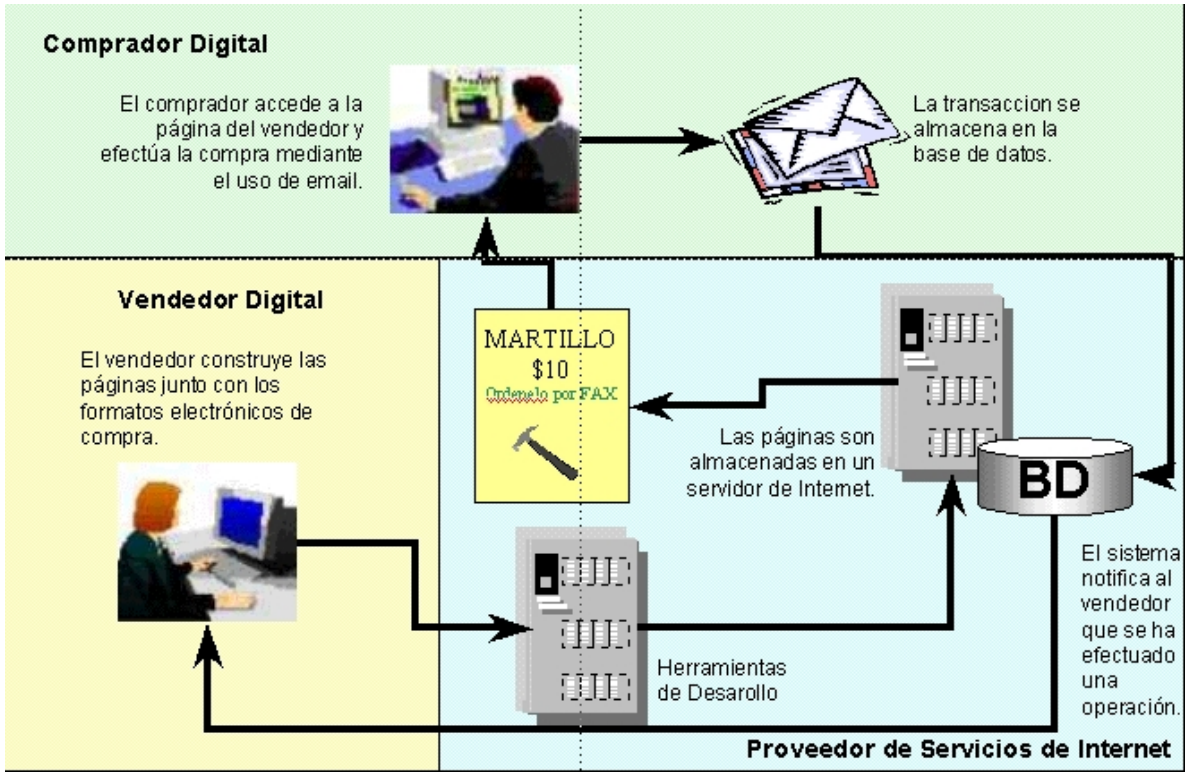


Figura 2-5. Modelo Híbrido de Hospedaje de Tienda Virtual

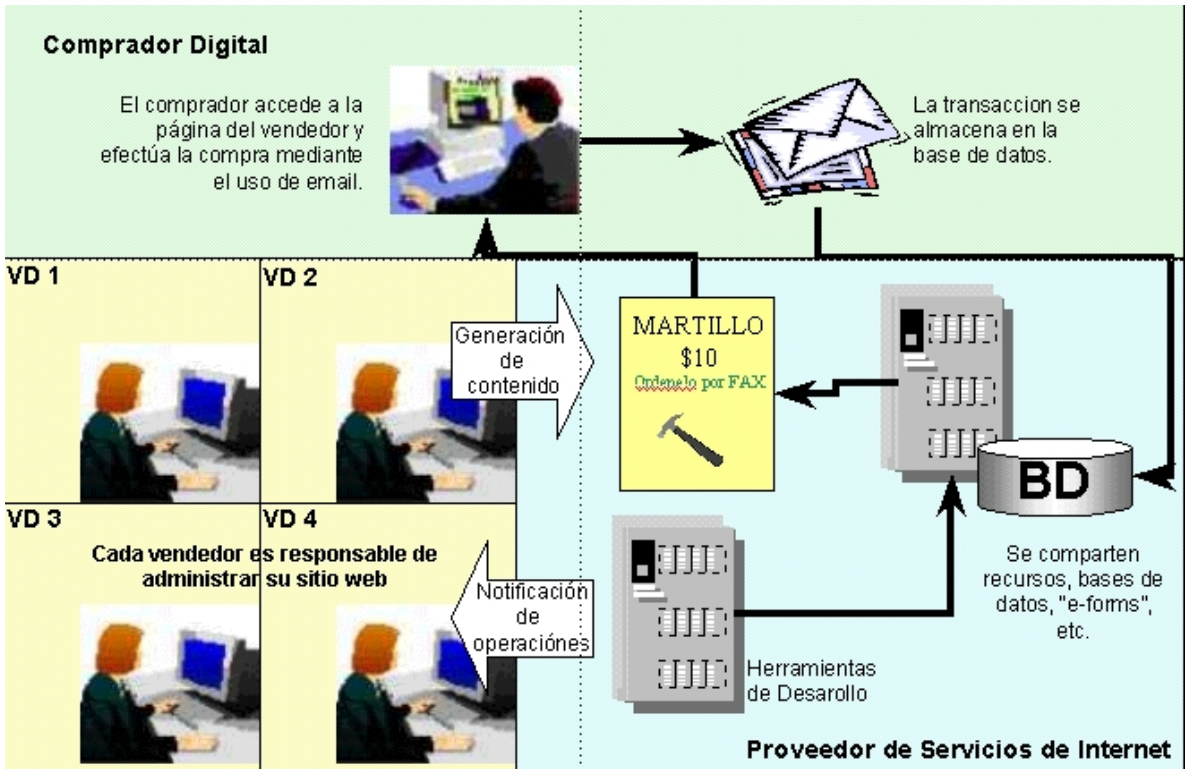


Figura 2-6. Modelo de Hospedaje para Nivel II de Comercio Electrónico

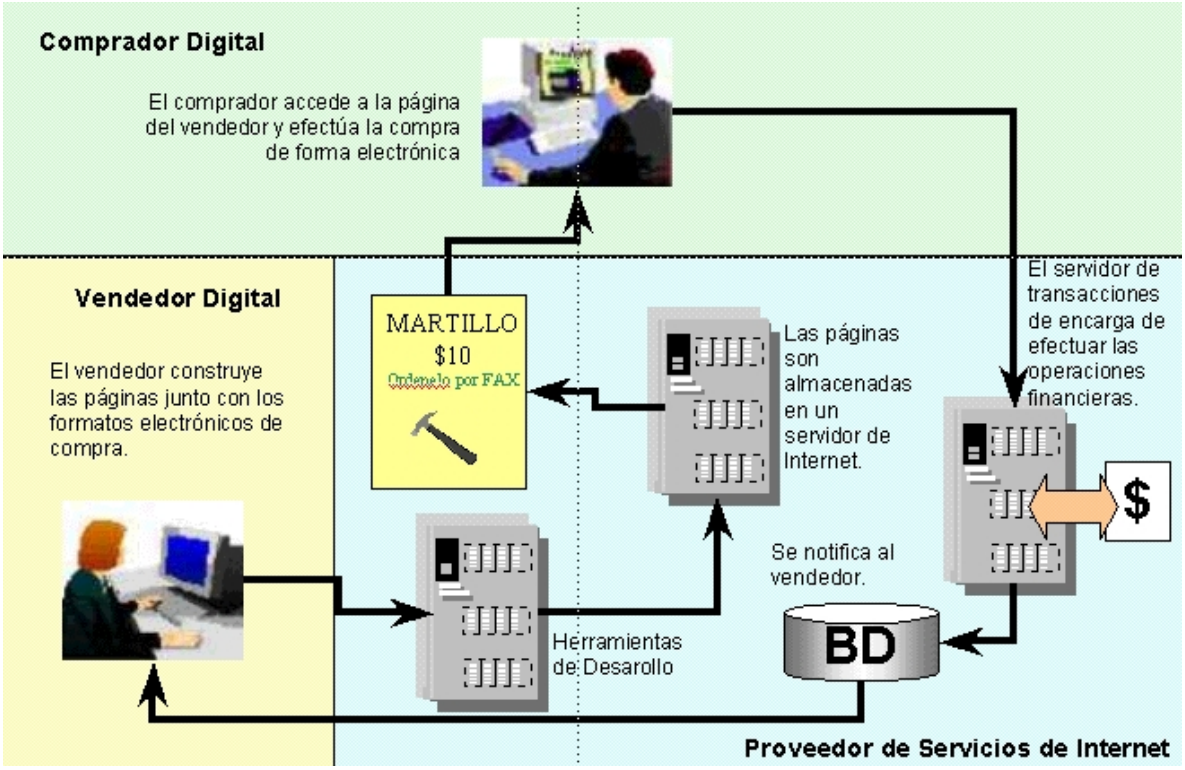


Figura 2-7. Modelo Avanzado de Hospedaje para Comercio Electrónico

2.5 Entes Activos en el Comercio Electrónico

Al desarrollarse el comercio electrónico se requiere de tener una buena infraestructura informática en la cual se determinen los principales entes activos a participar, ya que de acuerdo a lo que se implante se podrán crear modelos de operación entre los diferentes participantes, lo que se reflejaría en mayores ventajas para las partes involucradas.

2.5.1 Entes activos en el comercio electrónico de bienes entregados de forma electrónica

A continuación se presenta y define a los principales entes activos (crean, transmiten, reciben, procesan y almacenan información) dentro del comercio electrónico.

Figura 2-8. Representación de los entres participantes

2.5.1.1 Cliente o usuario final

El Cliente Final es el usuario principal del bien (consideraremos el caso de un *software*): lo compra, lo transfiere, lo almacena y lo procesa. Para obtenerlo establece diferentes relaciones con los otros entes, a través de los diferentes servicios de los cuales es usuario:

- Con el Banco: se registra, deposita fondos, permite la transferencia de dinero de sus cuentas a las de otros, puede bloquear su cuenta y pedir historial de actividades.
- Con la Tienda Virtual: se registra, selecciona, compra y transfiere el *software*; facturas digitales; y pide historial de actividades.

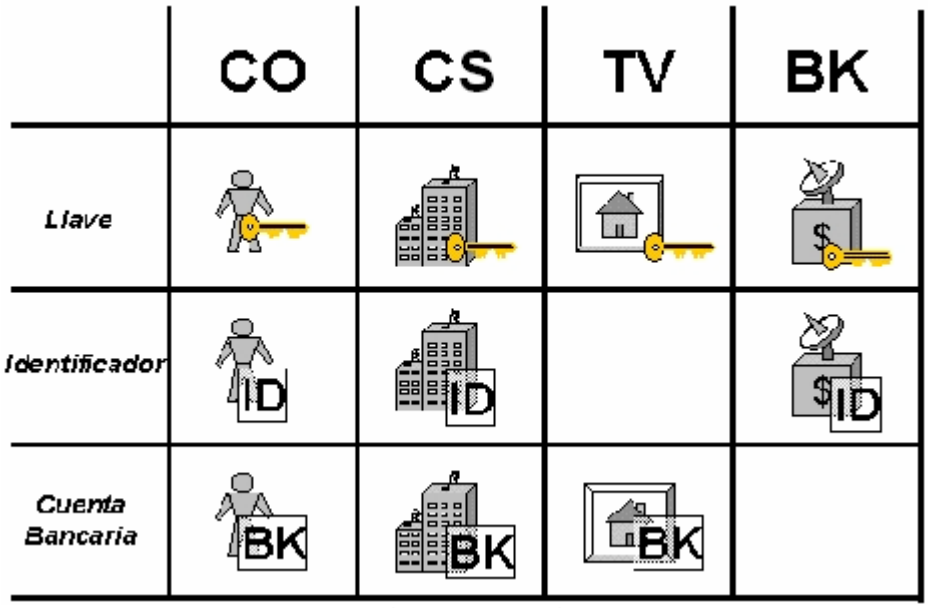


Figura 2-9. Representación de las características de cada ente

2.5.1.2 Banco

El Banco es el ente que maneja y resguarda el dinero de todos los demás entes en el modelo. Las transacciones entre diferentes bancos se hacen en un medio bancario seguro (sistemas cerrados, líneas privadas, etc.)

Los entes "Cliente Final", "Tienda Virtual" y "Casa de *Software*" son usuarios de sus servicios:

- Registro de usuario: Esto implica darle un número de cuenta, número de tarjeta y claves para realizar operaciones de banca electrónica y actividades comerciales por *Internet*.
- Realización de transferencias entre cuentas: Verificar las autorizaciones de transferencia, de compra y de saldo necesarias; para realizar los retiros, abonos y registro de actividades correspondientes.
- Proveer historial de actividades a cada usuario: Dar a cada usuario el registro actual de transacciones realizadas en su cuenta.

2.5.1.3 Casa de *software*

La Casa de *Software* es el ente que crea el *software*, pero no lo vende directamente al Cliente Final. Por ello necesita establecer relaciones con otros entes:

1. Con el Banco: Para darse de alta y tener una cuenta con la cual pueda realizar transacciones.
2. Con la Tienda Virtual: a quien le vende el *software*, por paquete o individual.

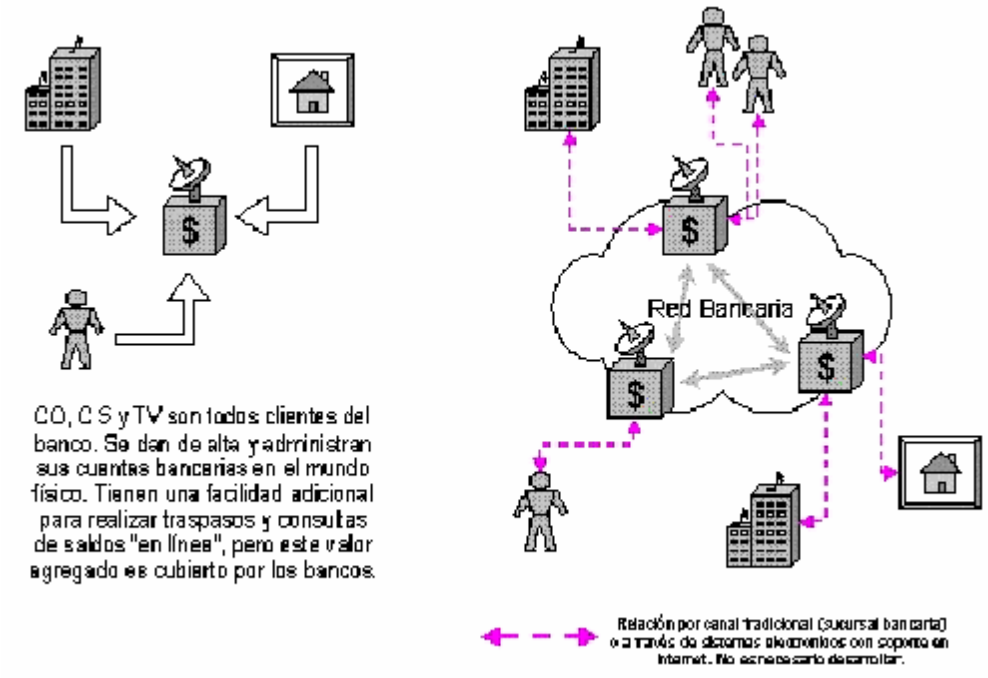


Figura 2-10. Relación básica entre entes

2.5.1.4 Tienda virtual

La Tienda Virtual vende el *software* a cada Cliente Final, personalizándolo de manera que únicamente el Cliente Final que la compro pueda ejecutar el *software*. Establece relaciones:

1. Con el Banco: Se da de alta para tener una cuenta con la cual pueda realizar transacciones.
2. Con la Casa de *Software*: Quien será su proveedor del material a vender.
3. Con el Cliente Final: A quien le venderá el producto.

Como se puede ver en la figura, cuando un comprador ha decido adquirir un *software*, y su tarjeta ha sido validada, la tienda virtual se encarga de determinar si el *producto* vendido se encuentra dentro de su base de datos, o si se deberá comunicar con el sistema de almacenamiento de la casa de *software*.

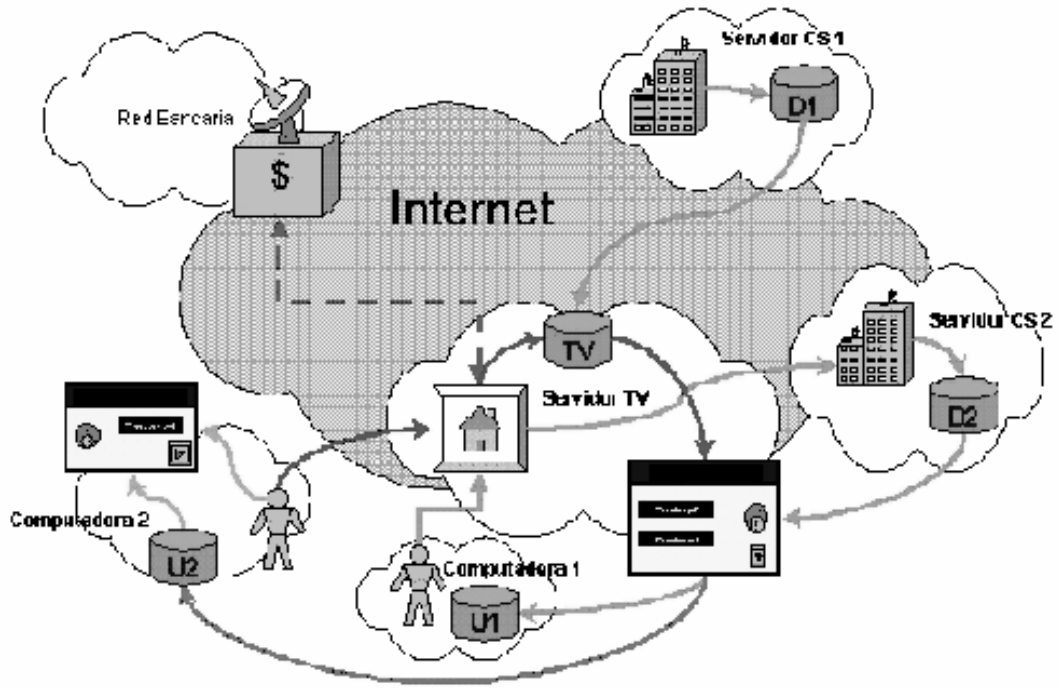


Figura 2-11. Mapa relacional de todo el sistema modelado

2.5.2 Entes activos en el comercio electrónico de bienes entregados por medios no electrónicos

Anteriormente se había comentado que para ser considerado comercio electrónico bajo esta modalidad se requería incluir el cobro del producto en línea, por lo que los entes que participan dentro de este modo son prácticamente los mismos que en el comercio electrónico entregados de forma electrónica, a excepción del desarrollador del *software* "Casa de *Software*" en su modalidad de venta electrónica, ya que en este ámbito entra cualquier "Empresa Elaboradora de Bienes Tangibles", así como empresas encargadas en la transportación de los bienes al usuario final "Servicio de Paquetería". A continuación

se describen las características que tienen estos dos “nuevos” entes activos involucrados en este medio.

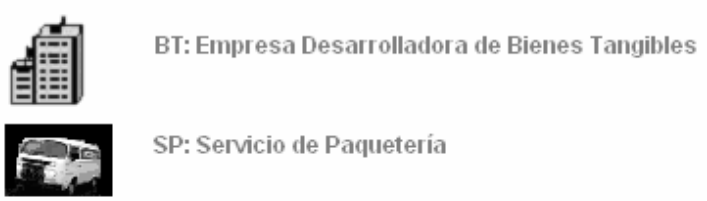


Figura 2-12. Entes activos

2.5.2.1 Empresa desarrolladora de bienes tangibles

La Empresa Desarrolladora de Bienes Tangibles es la encargada de crear el producto y que ha visto en el comercio electrónico la oportunidad de llegar a nuevos mercados que por su situación geográfica serían muy difíciles de llegar, es por ello que necesita establecer relaciones con otros entes:

1. Con el Banco: Para darse de alta y tener una cuenta con la cual pueda realizar transacciones.
2. Con la Tienda Virtual: a quien le promueve su catalogo de productos, para que esta lo publicite en su sitio *Web* y así poder llegar a la mayor cantidad de posibles compradores. Aunque esta elección es opcional, ya que la misma empresa puede tener su propio sitio *Web*.
3. Servicio de Paquetería: Es la que se encarga de hacer llegar la mercancía en el domicilio del destinatario.

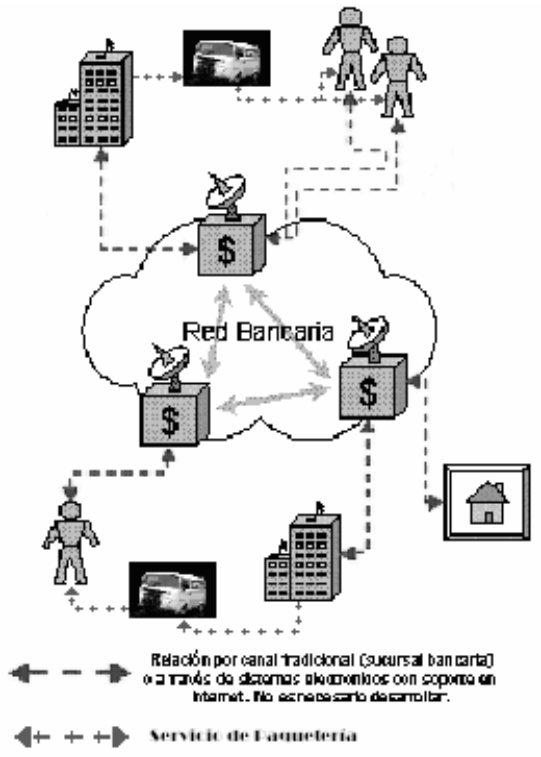


Figura 2-13. Relación de entes

2.5.2.2 Servicio de paquetería

El Servicio de Paquetería lleva el artículo comprado a cada Cliente Final, encargándose de que llegue en óptimas condiciones a la dirección pactada, así como entregar puntualmente el envío siempre y cuando las líneas transportistas hayan cumplido con sus horarios oficiales. Establece relaciones:

1. Con el Banco: Se da de alta para tener una cuenta con la cual pueda realizar transacciones; este ente puede ser descartado si el traslado lo cubre directamente el Cliente Final, de acuerdo a lo pactado en la modalidad de pago.
2. Con la Empresa Desarrolladora de Bienes: Quien será su proveedor del material a transportar.
3. Con el Cliente Final: A quien le llevara el producto, o inclusive le cobrara los costos de traslado del bien a entregar.

2.6 Modos de Operación entre los Participantes para el Comercio Electrónico

Una vez elegido el modelo de "hospedaje" a utilizar y definidos los entes activos a participar se puede definir el modo de operación por la cual se va a establecer la relación entre los diversos participantes involucrados directamente en el comercio electrónico. Por los diferentes procesos que se pueden realizar entre estos entes, se visualizan diversos modos de operación:

2.6.1 Modelo B2B y B2C

En este caso, el consumidor decide comprar un *software*, hace su solicitud y la Tienda Virtual envía la misma solicitud a la Casa de *Software* dueña del producto seleccionado; ésta valida la venta, lee el *software* de su base de datos, crea el *software* con su clave y las características del Usuario Final y lo envía a la Tienda Virtual. Al mismo tiempo, actualiza un cargo en sus bases de datos para que finalizado un período, o alcanzado un monto acordado, la Tienda Virtual deberá liquidar su adeudo con dicha firma.

Para la Tienda Virtual este escenario ofrece la ventaja de requerir moderados recursos en el almacenaje, soporte y distribución de la mercancía (pues solo debe manejar catálogos actualizados de las diferentes Casas de *Software*), no debe comprar los productos permanentemente (lo cual implicaría pagar por ellos una gran inversión), ya que toda la responsabilidad del manejo de la mercancía y su protección recae en cada Casa de *Software*.

El riesgo para la Tienda Virtual es el hecho de que solo actúa como un revendedor con poco, o nulo, valor agregado, lo cual implicaría que puede ser eliminado de la cadena productor-vendedor-consumidor pues cada Casa de *Software* puede establecer un sitio que hiciera la misma operación y tal vez ofreciera el producto a menor costo.

El riesgo para la Casa de *Software* es el posible fraude de que la Tienda Virtual aparente vender solo un *software* (a precio de usuario final), y de ahí lo utilice para revenderlo un sinnúmero de veces.

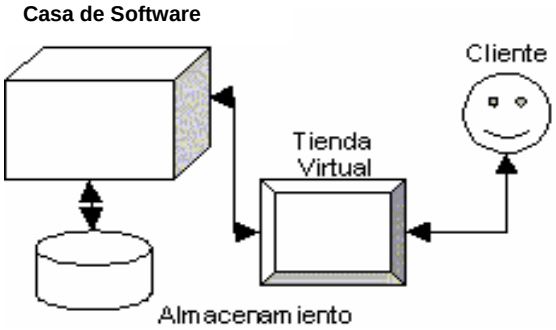


Figura 2-14. Modelo B2B y B2C

2.6.2 Modelo paquete y B2C

En este segundo caso, la Casa de *Software* y la Tienda Virtual acuerdan que la primera cede los derechos de un cierto paquete de *software* a cambio de una tarifa establecida entre ambas partes. En este caso, la responsabilidad de salvaguardar la integridad y seguridad del material es de la Tienda Virtual.

Las desventajas para la Tienda Virtual:

1. El hecho de poseer *software* que tal vez no le convenga por ser comprado en paquetes (siendo muy probable que dichos paquetes sean armados a criterio de la Casa de *Software*); y
2. El hecho de contar con los paquetes de manera permanentemente es razón por la cual cada Casa de *Software* los venderá a un precio elevado (considerando la popularidad del *software* en el momento de la venta).

Por consecuencia, la Tienda Virtual requerirá de un capital elevado para poder invertirlo en tener un inventario propio aceptable; y no todo el *software* adquirido garantiza el retorno de la inversión.

La oportunidad, o ventaja, es la libertad de efectuar las ventas como mejor le convenga (subir o bajar precios, crear promociones u ofertas, etcétera); y si el *software* aumenta su popularidad, aumentarán sus ventas.

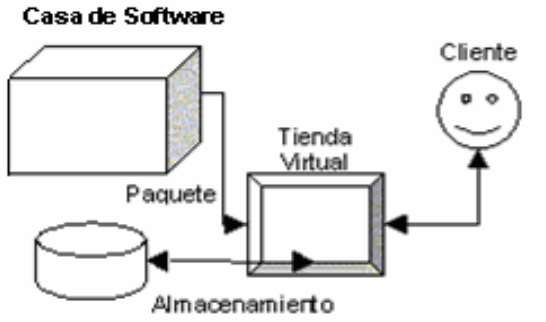


Figura 2-15. Modelo Paquete & B2C

2.6.3 Modelo regalía y B2C

La operación de este modelo se basa en que cada Casa de *Software* entrega su producto a la Tienda Virtual; y después de un lapso de tiempo, o alcanzada una determinada cifra de ventas, la segunda paga las regalías correspondientes a cada firma.

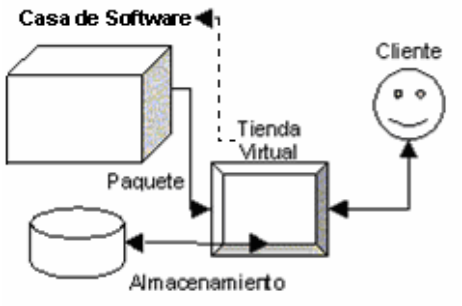


Figura 2-16. Modelo Regalía & B2C

La ventaja para la Tienda Virtual es que no debe invertir gran cantidad de capital, pues en principio no debe invertir en mercancía. Pero la gran disyuntiva para establecer este escenario es el hecho de que se debe plantear todo un sistema de seguridad para asegurar que la Tienda Virtual pague a cada Casa de *Software* lo que en verdad vende, es decir, que deben existir las herramientas suficientes para que no se pueda negar lo que se ha facturando. Esto se puede acentuar por el inconveniente de que de antemano se establece un ambiente de desconfianza entre la Casa de *Software* y la Tienda Virtual. Aunque el diseño del esquema de seguridad necesario para establecer este escenario es posible, el análisis y diseño de las herramientas correspondientes pueden ser motivo para desarrollar una tesis únicamente enfocada a este problema.

2.6.4 Modelo general

Este consiste en un modelo combinado. La idea básica es que la Tienda Virtual compre permanentemente el *software* más popular (según encuestas de *internet*, revistas) pero maneje los catálogos completos de todo el *software* que tiene cada Casa de *Software*. Cuando el Usuario Final solicita algún producto que no se encuentre en su inventario, la Tienda Virtual se encarga de gestionar la compra de dicho producto directamente con la Casa de *Software* correspondiente, descargarlo y luego generar el *software* a vender al Usuario Final.

Este escenario plantea la desventaja para la Tienda Virtual de requerir un capital moderado para adquirir el *software* más popular, pero con las ventajas del segundo escenario analizado anteriormente. Y para la Casa de *Software* establece la ventaja de que no debe preocuparse de algún posible fraude como en el primer y tercer escenarios planteados.

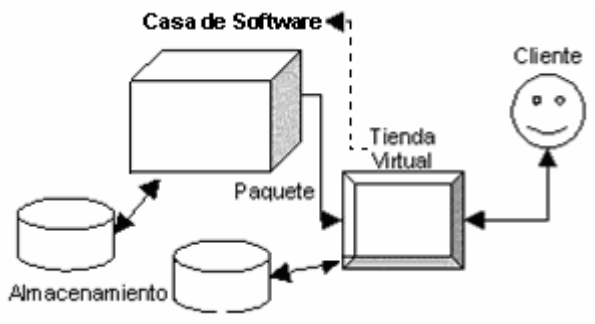


Figura 2-17. Modelo General

Existen muchas otras variaciones respecto al mecanismo de venta, sin embargo, todas ellas pueden ser englobadas dentro de alguna de las categorías descritas.

2.7 Necesidad de la Seguridad en el Comercio Electrónico

El desarrollo de la tecnología ha llegado a afectar tan profundamente al mundo y se ha integrado de tal modo en la actividad humana, que ha dejado de ser una finalidad aislada. De esta forma, cuando los avances tecnológicos son asimilados socialmente, se generan nuevas necesidades y problemas [10].

El mercado de la información hace posible que todo el mundo compre, venda e intercambie bienes y servicios sin tener forzosamente que registrarse, ni ser controlado por alguna autoridad central omnipresente y omnipotente.

El problema central de la seguridad informática en el mercado de la información reside en:

1. La gran conectividad de *Internet*, y
2. La digitalización de la información y el dinero.

La gran abundancia de conexiones posibles en la infraestructura de *Internet* permite a unos acceder electrónicamente a la información de otros (negocio, consumidor o gobierno) con intenciones sospechosas y posibles consecuencias desastrosas [10].

Como es sabido, *Internet* es un medio en el cual se expande rápidamente cualquier error, o atentado, y con severas repercusiones, en el comercio electrónico (falla en la actualización de precios, cambio no autorizado en los mismos, cambio del destinatario en los pedidos, obtención de los números de tarjetas de crédito en forma ilegal para realizar delitos financieros, etc.) [11].

Con el uso de los procesos digitales, es enorme el crecimiento del volumen de información que los gobiernos, los competidores, los delincuentes o gente simplemente entrometida es capaz de interceptar. Es tan grande el volumen y el alcance de la información en formato electrónico que su posible violación por individuos no autorizados, debe ser tomada en cuenta por individuos, negocios y el gobierno [10].

2.7.1 Herramientas para transacciones seguras

La manera como se efectúan los pagos en las transacciones electrónicas y en general como se maneja el dinero en el mercado de la información; permite la realización de delitos en magnitud (cifras), alcance (sin fronteras físicas) y cantidad (muchos y muy diversos usuarios) bastante grandes [10].

En esta cadena comercial de valor, el eslabón más débil ha sido, y todavía es, la forma de pago, consiste también en el mayor obstáculo tanto técnico, como psicológico, que debe ser vencido para que se produzca el despegue definitivo del comercio electrónico. Mientras no exista confianza, mientras los usuarios teman al fraude, mientras se desconozcan los sistemas de pago empleados y su fiabilidad, será difícil que se observe un incremento sustancial en esta forma de comercio.

Sin embargo, en los últimos años ha ido surgiendo un número considerable de tecnologías y sistemas de pago electrónico que ofrecen las garantías de seguridad e integridad necesarias para realizar las compras en línea de una manera fiable y sin sorpresas. La piedra angular de todas ellas es la criptografía, que proporciona los mecanismos necesarios para asegurar la confidencialidad, e integridad, de las transacciones. Como se verá a continuación:

Del conjunto de técnicas existentes en el mercado, cabe destacar dos tipos de sistemas: SSL y SET. Mientras que SSL ofrece un nivel aceptable de seguridad en las compras por *Internet*, pues garantiza que la información que se transmite viaje de forma cifrada, SET ofrece un nivel de seguridad óptimo ya que, además, permite las identificaciones unívocas de las partes (comprador, vendedor, etcétera) involucradas en la transacción. A continuación se verá en qué consisten estos protocolos y cómo proporcionan la seguridad requerida aunque será hasta el capitulo siguiente donde se abordarán de forma más extensa los mecanismos criptográficos empleados.

Independientemente del sistema de seguridad implementado, se debe tener la certeza de que un comercio es seguro cuando se den las siguientes condiciones:

1. Hay un candado cerrado (o una llave) en la parte inferior del navegador
2. La dirección de la página comienza por https:

2.7.1.1 SSL

SSL (*Secure Sockets Layer*) fue diseñado y propuesto en 1994 por *Netscape Communications Corporation* junto con su primera versión de Navegador. Sin embargo, no fue hasta su tercera versión, conocida como SSL v3.0 que alcanzó su madurez, superando los problemas de seguridad y limitaciones de sus predecesores. En su estado actual, proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP. En la actualidad, la mayoría de los comercios que venden a través de *Internet* cuentan con un sistema que utiliza el protocolo SSL. Para que resulte operativo, el consumidor sólo precisa disponer de un navegador (*Internet* Explorer, *Netscape Communicator*, etc.). En el comercio deberá instalarse un certificado en el servidor donde se tenga alojado el sitio "*Web*", certificado que podrá ser obtenido a través de su Entidad Financiera, o de una Autoridad de Certificación acreditada.

Como funciona SSL

El rasgo que distingue a SSL de otros protocolos para comunicaciones seguras, como el hoy prácticamente extinto S-HTTP, es que se ubica en la pila OSI entre los niveles de transporte (TCP/IP) y de aplicación (donde se encuentran los conocidos protocolos HTTP para *Web*, FTP para transferencia de ficheros, SMTP para correo electrónico, Telnet para conexión a máquinas remotas, etc.). Gracias a esta característica, SSL resulta muy flexible, ya que puede servir para asegurar potencialmente otros servicios además de HTTP para *Web*, sin más que hacer pequeñas modificaciones en el programa que utilice el protocolo de transporte de datos TCP.

SSL proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, que puede elegirse entre DES, triple-DES, RC2, RC4 o IDEA, y cifrando la clave de sesión de los algoritmos anteriores mediante un algoritmo de cifrado de clave pública, típicamente el RSA (algoritmos analizados en el capitulo siguiente). La clave de sesión es la que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea descubierta por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5, o SHA, se pueden usar como algoritmos de huella digital (*hash*). Esta posibilidad de elegir entre tan amplia variedad de algoritmos dota a SSL de una gran flexibilidad criptográfica.

Durante el protocolo SSL, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad; durante la sesión segura abierta, SSL proporciona un canal de comunicaciones seguro entre los servidores *Web* y los clientes (los navegadores) a través del cual se intercambiará cifrada la información relevante, como el URL y los contenidos del documento solicitado, los contenidos de cualquier formulario enviado desde el navegador, las *cookies* enviadas desde el navegador al servidor y viceversa y los contenidos de las cabeceras HTTP.

SSL constituye la solución de seguridad implantada en la mayoría de los servidores *Web* que ofrecen servicios de comercio electrónico. Su mayor mérito radica en ofrecer respuesta al principal problema que afronta el comercio en línea: la renuencia de los usuarios a enviar su número de tarjeta de crédito a través de un formulario *Web* por el temor de que caiga en manos de un *hacker* y por la desconfianza generalizada con respecto a *Internet*.

2.7.1.2 SET

SET, o Transacciones Electrónicas Seguras (*Secure Electronic Transaction*) es un protocolo estandarizado y respaldado por la industria, diseñado para salvaguardar las compras pagadas con tarjeta a través de redes abiertas, incluyendo *Internet*. El estándar SET fue desarrollado en 1995 por Visa y MasterCard, con la colaboración de otras compañías líderes en el mercado de las tecnologías de la información, como Microsoft, IBM, Netscape, RSA, VeriSign y otras.

En 1997 Visa y MasterCard formaron SET LLC (comúnmente conocida como "SETCo") para que implantase la especificación. En cuanto el protocolo SET 1.0 fue finalizado, comenzó a emerger una infraestructura basada en el mismo para dar soporte a su uso a gran escala. Ya existen numerosos fabricantes de *software* que han empezado a crear

productos para consumidores y comerciantes que deseen realizar sus compras de manera segura disfrutando de las ventajas ofrecidas por SET.

- Asegura la confidencialidad, e integridad, de la información transmitida.
- Permite la autenticación de los compradores, vendedores y Entidades Financieras involucradas en la transacción.
- Garantiza el no rechazo de las operaciones realizadas.

Su empleo requiere que cada uno de los participantes en la transacción disponga de un certificado SET, así como de un "*software*" específico. Estos elementos se podrán conseguir a través de una Entidad Financiera:

- El comprador dispondrá de un aplicativo, denominado "*Electronic Wallet*" o "Cartera Electrónica", en el cual dará de alta las tarjetas con las que desee realizar pagos: cada una de estas tarjetas se asociará en el proceso de alta de un certificado.
- El vendedor dispondrá de un aplicativo, denominado "*Merchant Software*" o "Programa Gestor", que se instalará en la "*Web*" del comercio y gestionará las operaciones de compra bajo el protocolo SET.

Existen además otros componentes en las operaciones de compra realizadas con SET:

- Pasarela de Pagos ("*Payment Gateway*"): es un sistema de comunicaciones que permite procesar y autorizar las transacciones de pago con tarjeta.
- Autoridad de Certificación: tercera parte confiable que, a través de las Entidades Financieras, emite certificados SET.
- Emisor: Entidad Financiera emisora de la tarjeta del comprador.
- Adquirente: Entidad Financiera con quien trabaja el comercio, para la solicitud y liquidación de los pagos.

El protocolo definido por SET especifica el formato de los mensajes, las codificaciones y las operaciones criptográficas que deben usarse. No requiere un método particular de transporte, de manera que los mensajes SET pueden transportarse sobre HTTP en aplicaciones *Web*, sobre correo electrónico, o cualquier otro método. Como los mensajes no necesitan transmitirse en tiempo real, son posibles implantaciones de SET eficientes basadas en correo electrónico u otros sistemas asíncronos.

En su estado actual SET solamente soporta transacciones con tarjeta de crédito/débito, y no con tarjetas monedero. Se está trabajando en esta línea para extender el estándar de manera que acepte nuevas formas de pago. Al mismo tiempo se están desarrollando proyectos para incluir los certificados SET en las tarjetas inteligentes, de tal forma que el futuro cambio de tarjetas de crédito a tarjetas inteligentes pueda incorporar el estándar SET. [43]

2.7.2 Precauciones a considerar

Es indudable que siempre habrá tentativas de delito en el mercado de la información (*Internet* es un medio en el cual se refleja y se extiende la naturaleza humana) y que algunos de dichos delitos nunca se detectarán [10].

Las tecnologías que producen una buena seguridad (criptografía y contrainteligencia) también producen buenos ataques (criptoanálisis e inteligencia). Todos los participantes en el comercio electrónico se involucrarán en una serie de medidas, y contramedidas para obtener alguna ventaja en el conocimiento del otro y proteger el propio.

Con independencia de la dificultad matemática para quebrantarlo, todo esquema criptográfico es vulnerable a otros tipos de ataques. En primer lugar, los violadores pueden infiltrarse en nuestro campo bajo la forma de aliados y comprometer las maneras de hacer, compartir y disponer de los códigos, al poner todo eso en conocimiento del enemigo. Lo más destructivo es que una persona a la que se ha confiado la empresa criptográfica sea sobornada por el campo enemigo [10].

En consecuencia el factor humano sigue siendo un eslabón débil en la cadena de la seguridad informática y habrá que tomarlo en cuenta [10].

CAPÍTULO 3

CRIPTOGRAFÍA Y SEGURIDAD DE LA INFORMACIÓN EN REDES DE DATOS

Criptografía y Seguridad de la Información en Redes de Datos

Los primeros conceptos de seguridad se tienen registrados en los inicios de la escritura con los Sumerios (3000 AC). También en escritos antiguos como la Biblia, y obras de Homero, Cicerón, Cesar muestran ciertos rasgos de la seguridad en las guerras y en los gobiernos. Los descubrimientos arqueológicos marcan, sin duda, las más importantes pruebas de seguridad de las culturas antiguas por ejemplo, las pirámides egipcias.

Sin embargo, a partir del siglo XVIII, los descubrimientos científicos y el conocimiento resultante de la imprenta han contribuido a la cultura de la seguridad. Las teorías de probabilidad, predicción y reducción de fallos y pérdidas han contribuido a los sistemas de seguridad.

La seguridad moderna se originó con la Revolución Industrial para combatir los delitos y movimientos laborales, tan comunes en aquella época. Finalmente, un teórico y pionero de la administración, Henry Farol [24], identifica en 1919 la Seguridad como una de las funciones empresariales, luego de la técnica, comercial, financiera, contable y directiva.

En cambio desde el punto de vista técnico, la seguridad está en manos de la dirección de las organizaciones y, en última instancia, en cada uno de nosotros y en nuestro grado de concientización respecto a la importancia de la información y del conocimiento en este nuevo milenio.

Este problema será solucionado satisfaciendo las necesidades de comprensión del concepto de "Seguridad" y "Sistema Informático" en tomo de algo (organización o particular) que gestiona información. Para esto es necesario adaptar los principios de Seguridad expuestos en un contexto informático y viceversa. En definitiva los expertos en seguridad y los expertos en informática deben interactuar interdisciplinariamente para que realmente exista Seguridad Informática.

En el presente documento, cada vez que se mencione información se estará haciendo referencia a la información que es procesada por un sistema informático; definiendo este último como el "conjunto formado por las personas, computadoras (*hardware y software*), papeles, medios de almacenamiento digital, el entorno donde actúan y sus interacciones" [1].

Luego:

"El objetivo de la seguridad informática será mantener la Integridad, Disponibilidad, Privacidad (sus aspectos fundamentales), Control y Autenticidad de la información manejada por computadora." [1]

Contrariamente a lo que se piensa, este concepto no es nuevo y técnicamente se originó con los grandes centros de cómputo, con el pasar de los años y con el incremento de las computadoras personales y de las redes de datos, la seguridad de la información se ha incrementado aunque todavía falta mucho por hacer ya que con el desarrollo de nuevas tecnologías informáticas también se ha incrementado el campo de acción de conductas antisociales y criminales que anteriormente eran imposibles de efectuar.

Para evitar las amenazas a la seguridad del sistema se puede contar con una serie de medidas para proteger los sistemas de procesamiento de datos y de transferencia de información de una organización. Estas medidas hacen uso de uno o varios mecanismos de seguridad, aunque la mayoría de estos se basa en el uso de técnicas criptográficas basadas en el cifrado de la información, es por tal motivo que en este capitulo se describirá en términos generales los aspectos de seguridad, de mucho interés actual, empleados en la elaboración y protección de un sitio de comercio electrónico.

3.1 Introducción a la Seguridad de la Información

El concepto de seguridad en la información es mucho más amplio que la simple protección de los datos a nivel lógico. Para alcanzar una seguridad real se tiene que tener en cuenta múltiples factores, tanto internos como externos. En primer lugar habría que caracterizar el sistema que va a albergar la información para poder identificar las amenazas, y en este sentido se podría hacer la siguiente subdivisión:

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

- *Sistemas interconectados*. En la actualidad casi cualquier computadora esta integrada a alguna red, enviando y recibiendo información del exterior casi constantemente. Esto hace que las redes de datos sean cada vez más complejas y conlleven un riesgo potencial que no puede en ningún caso ser ignorado.

En cuanto a los tipos de seguridad que se han identificado se pueden clasificar de la siguiente forma:

- *Seguridad física*. Se engloba dentro de esta categoría a todos los asuntos relacionados con la salvaguarda de los soportes físicos de la información, más que de la información propiamente dicha. En este nivel estarían, entre otras, las medidas contra incendios y sobrecargas eléctricas, las políticas de *backup*, etc.

  También se suelen tener en cuenta dentro de este punto aspectos relacionados con la restricción de acceso físico a las computadoras únicamente a personas autorizadas.

- *Seguridad de la información.* En esta categoría se presta atención a la preservación de la información frente a usuarios no autorizados. Para ello se puede emplear tanto la llamada criptografía simétrica, como asimétrica.

- *Seguridad del canal de comunicación*. Los canales de comunicación rara vez se consideran seguros. Debido a que en la mayoría de los casos escapan de control, ya que pertenecen a terceros, resulta imposible asegurarse totalmente de que no están siendo escuchados, o intervenidos.

- *Problemas de autentificación.* Debido a los problemas del canal de comunicación, es necesario asegurarse de que la información que se recibe en la computadora

viene de quien realmente se cree que viene. Para esto se suele emplear criptografía asimétrica, en conjunción con funciones llamadas de huella (*hash*).

- *Problemas de suplantación.* En las redes se tiene el problema añadido de que cualquier usuario autorizado puede acceder al sistema desde fuera, por lo que se debe de confiar en sistemas fiables para garantizar que los usuarios no están siendo suplantados por intrusos. Normalmente se emplean mecanismos basados en contraseñas para conseguir esto.

- *No repudio.* Cuando se recibe un mensaje no solo es necesario poder identificar de forma unívoca al remitente, sino que éste asuma todas las responsabilidades derivadas de la información que haya podido enviar. En este sentido es fundamental impedir que el emisor pueda "repudiar" un mensaje, es decir, negar su autoría sobre el.

3.1.1 Objetivo de la Seguridad de la Información

Establecer el costo de la información es algo relativo, debido a que es un recurso que en la mayoría de los casos, no se valora adecuadamente debido a su intangibilidad, cosa que no ocurre con los equipos, las aplicaciones y la documentación.

No obstante hay Información que debe, o puede, ser pública: puede ser visualizada por cualquier persona; y por otro lado aquella que debe ser privada: sólo puede ser visualizada por un conjunto de personas que se ocupa de ella. En esta última categoría es donde se tiene que tener mucho cuidado, y es el objetivo de la seguridad informática, ya que se tiene que preservar reconociendo las siguientes características intrínsecas en la información. Se puede decir que esta es:

1. Crítica: es indispensable para garantizar la continuidad operativa.
2. Valiosa: es un activo con valor en sí misma.
3. Sensitiva: debe ser conocida por las personas que la procesan y sólo por ellas.

Para hacer frente a las amenazas contra la seguridad del sistema, se define una serie de servicios para proteger los sistemas de procesamiento de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno, o varios mecanismos de seguridad. Una clasificación útil de los servicios de seguridad es la siguiente:

- *Confidencialidad, o Privacidad:* requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas, o tal vez sólo a porciones o segmentos seleccionados de los datos, por ejemplo mediante cifrado. La confidencialidad del flujo de datos protege la identidad del origen y destino(s) del mensaje. En casos de falta de confidencialidad, la información puede provocar severos daños a los usuarios autorizados, o volverse obsoleta.

- *Autenticación:* requiere una identificación correcta del origen del mensaje, asegurando que la entidad no es falsa. Se distinguen dos tipos: una llamada de entidad, que asegura la identidad de las entidades participantes en la comunicación, mediante alguna biométrica (huellas dactilares, identificación de

iris, etc.), tarjetas de banda magnética, contraseñas, o procedimientos similares; y otra llamada de origen de información, que asegura que una unidad de información proviene de cierta entidad, siendo la firma digital el mecanismo más extendido.

- *Integridad:* requiere que la información sólo pueda ser modificada por las entidades autorizadas. Esta modificación puede incluir: escritura, cambio, borrado y creación de mensajes transmitidos. La integridad de datos asegura que los datos recibidos no han sido modificados de ninguna manera, utilizando por ejemplo una función "huella" (hash) criptográfica con firma. Mientras que la integridad de secuencia de datos asegura que la secuencia de los bloques o unidades de datos recibidos, no hayan sido alterados y que no haya unidades repetidas, o perdidas.

- *No repudio:* ofrece protección a un usuario frente a la posibilidad de que otro usuario niegue posteriormente que en realidad se realizó cierta comunicación. Esta protección se efectúa por medio de una colección de evidencias lógicas irrefutables que permitirán la resolución legal de cualquier disputa. El no repudio de origen protege al receptor de que el emisor niegue haber enviado el mensaje. Mientras que el no repudio de recepción protege al emisor de que el receptor niegue haber recibido el mensaje. Las firmas digitales constituyen el mecanismo más empleado para este fin.

- *Control de acceso:* requiere que el acceso a los recursos (información, capacidad de cálculo, nodos de comunicaciones, entidades físicas, etc.) sea controlado y limitado por el sistema destino, mediante el uso de contraseñas, o claves *hardware*.

- *Disponibilidad:* capacidad de mantener la información siempre disponible para ser procesada por las personas autorizadas. Esto requiere que la misma se mantenga correctamente almacenada con el *hardware* y el *software* funcionando perfectamente, y que se respeten los formatos para su recuperación en forma satisfactoria.

3.1.2 Mecanismos de seguridad

Para proporcionar los servicios de seguridad citados, es necesario incorporar en los niveles adecuados del modelo de referencia OSI para redes, los siguientes mecanismos de seguridad [21]:

1. Cifrado: el cifrado puede hacerse mediante el uso de criptosistemas simétricos, o asimétricos y puede aplicarse extremo a extremo, o a cada enlace del sistema de comunicaciones. El mecanismo de cifrado proporciona el servicio de confidencialidad de los datos y puede complementarse con otros mecanismos para conseguir diversos servicios de seguridad.

2. Firmado digital: la firma digital se puede definir como un conjunto de datos que se añaden a una unidad de datos de modo que protejan a ésta contra cualquier falsificación, permitiendo al receptor comprobar el origen y la integridad de los

mismos. Para ello, se cifra la unidad de datos junto con alguna componente secreta del firmante, y se obtiene un valor de control ligado al resultado cifrado. El mecanismo de cifrado digital aporta los servicios de integridad de los datos, autentificación del emisor y no repudio con prueba de origen. Para que se pueda proporcionar el servicio de no repudio con prueba de entrega, hay que forzar al receptor para que envíe un acuse de recibo firmado digitalmente.

3. Control de acceso: se usa para verificar la capacidad de un ente para acceder a un recurso dado. El control de acceso se puede llevar a cabo en el origen, o en un punto intermedio, y se encarga de asegurar que el emisor está autorizado a comunicarse con el receptor, o a usar los recursos de comunicación.

4. Integridad de datos: hay que distinguir entre la integridad de una unidad de datos individual y la integridad de una secuencia de unidades de datos. Para lograr integridad de una unidad de datos, el emisor añade datos suplementarios a la unidad de datos. Estos datos suplementarios se obtienen en función de la unidad de datos y, generalmente, se cifran. El receptor genera los mismos datos suplementarios a partir de la unidad original y los compara con los recibidos. Para proporcionar integridad para una secuencia de unidades de datos se requiere, adicionalmente, algún mecanismo de ordenación, tal como el uso de números de secuencia, un sello temporal, o un encadenamiento criptográfico entre las unidades.

5. Intercambio de autentificación, que tiene los grados siguientes:
    - Autentificación simple: el emisor envía su identificador y una contraseña al receptor, el cual los comprueba.
    - Autentificación fuerte: utiliza propiedades de los criptosistemas de clave pública. Un usuario se autentifica mediante su identificador y su clave privada. Su interlocutor debe verificar que aquel, efectivamente, posee la clave privada, para lo cual debe obtener, de algún modo, la clave pública del primero. Para ello deberá obtener su certificado. Un certificado es un documento firmado por una Autoridad de Certificación (una tercera parte de confianza) y válido durante el periodo de tiempo determinado, que asocia una clave pública a un usuario.

3.1.3 Amenazas a la seguridad en redes de datos

La seguridad en la comunicación a través de redes, especialmente *Internet*, consistente en prevenir, impedir, detectar y corregir violaciones a la seguridad durante la transmisión de información, además de la seguridad de los servidores, que comprende la seguridad de sistemas operativos y bases de datos. Considerando la información esencialmente en forma digital la protección se asegurará principalmente mediante medios lógicos, envés de físicos.

Las amenazas a la seguridad en el comercio electrónico pueden caracterizarse modelando el sistema como un flujo de información desde una fuente (servidor de la tienda virtual), como por ejemplo una canción, o clave de tarjeta bancaria; a un destino, como por ejemplo el usuario final. Un ataque es la realización de una amenaza.

Las cuatro categorías generales de amenazas, o ataques, son las siguientes:

1. *Interrupción:* un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son la destrucción de un elemento *hardware*, como un disco duro, cortar una línea de comunicación, o deshabilitar el sistema de gestión de archivos.

2. *Intercepción:* una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa, o una computadora. Ejemplos de este ataque consisten en introducirse en una línea para procurarse datos que circulen por la red y la copia ilícita de archivos, o programas (intercepción de datos), o bien la lectura de las cabeceras de paquetes para descubrir la identidad de uno, o más de los usuarios implicados en la comunicación (intercepción de identidad).

3. *Modificación:* una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son el cambio de valores en un archivo de datos, alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transmitidos por la red.

4. *Generación:* una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son la inserción de mensajes espurios en una red, o añadir registros a un archivo.

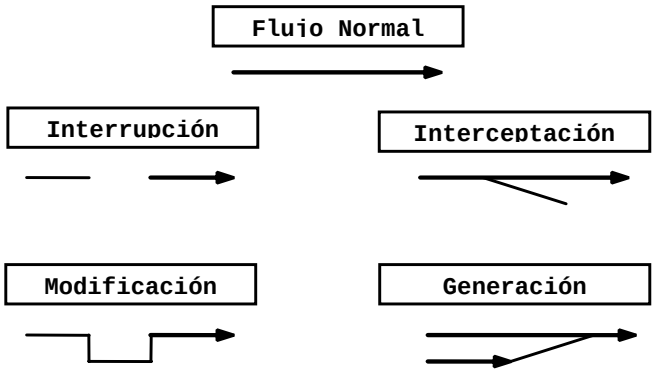


Figura 3-1. Representación de ataques en la comunicación

Las amenazas pueden ser analizadas en tres momentos: antes del ataque, durante y después del mismo. Estos mecanismos conformarán políticas que garantizarán la seguridad de nuestro sistema informático.

- *La prevención:* mecanismos que aumentan la seguridad de un sistema durante su funcionamiento normal. Por ejemplo el cifrado de información para su posterior transmisión.

- *La detección:* mecanismos orientados a revelar violaciones a la seguridad. Generalmente son programas de auditoria.

- *La* recuperación: mecanismos que se aplican, cuando la violación del sistema ya se ha detectado, para retomar éste a su funcionamiento normal. Por ejemplo recuperación desde las copias de seguridad (*backup*) realizadas.

3.2 Introducción a la Criptografía

La criptografía, en el contexto de redes informáticas, es una parte de la seguridad que estudia los métodos y procedimientos, mediante algoritmos matemáticos, para modificar los datos de tal manera que solamente los usuarios que tengan la clave adecuada puedan tener acceso a la versión original de los mismos, asegurar que estos datos no fueron modificados entre el remitente y el destinatario.

Actualmente, la criptografía involucra varias formas de encriptación/desencriptación, así como diferentes métodos de autenticación. Aunque sus métodos y aplicaciones siguen siendo cada vez más complejos, la criptografía sigue girando fundamentalmente alrededor de problemas matemáticos difíciles de solucionar. Un problema puede ser difícil de resolver porque su solución requiere de cierto conocimiento secreto, como la clave para desencriptar un mensaje cifrado, o para firmar un documento digital. También puede ser que sea intrínsecamente difícil de solucionar, en términos de los requerimientos matemáticos, o de cómputo, necesarios para decodificar el mensaje encriptado.

Por su parte, el área que se ocupa del estudio sistemático de los métodos para desencriptar información encriptada se denomina *criptoanálisis*, y es practicada por los *criptoanalistas*.

3.2.1 Objetivos de la criptografía

De todos los objetivos de seguridad informática, los siguientes cuatro se basan en algoritmos criptográficos y forman un soporte básico del cual los otros se pueden derivar, y son:

Confidencialidad o Privacidad
Integridad de los datos
Autenticidad
No repudio

Tabla 3-1: Objetivos Criptográficos [14,4]

Sistemas de Encriptado. Es un término general para referirse a un conjunto de primitivas usadas para proveer servicios de seguridad informática. Los requerimientos para los sistemas de encriptado fueron establecidos por Kerckhoff en 1883 y actualmente siguen siendo útiles para el diseño de estos sistemas, dichos requerimientos son [16]:

1. El sistema debe ser inquebrantable en la práctica, aunque en teoría no lo sea.

2. La revelación de los detalles del sistema no debe ser un inconveniente para las partes involucradas.

3. La clave o claves utilizadas deben ser recordadas sin registrarlas en papel (un medio inseguro) y su cambio no debe presentar ningún problema.

4. El criptograma, o texto cifrado, debe ser transmisible por un medio de telecomunicaciones.

5. El aparato de encriptación (implementación) debe ser portátil y operable por cualquier persona.

6. El sistema debe ser fácil de utilizar, sin requerir el conocimiento de una larga lista de reglas ni gran capacidad mental.

3.2.2 Comunicaciones seguras sobre redes inseguras

El crecimiento exponencial de los usuarios y organizaciones conectadas a *Internet* ha originado el tránsito a través de ella de informaciones de todo tipo, desde noticias y correos electrónicos, hasta complejas transacciones que requieren medidas específicas de seguridad que garanticen la confidencialidad, la integridad y constaten el origen de los datos.

En una comunicación entre dos máquinas, se supone la existencia de un emisor y un receptor, los cuales quieren intercambiar mensajes. El posible enemigo que quiere interferir de algún modo la comunicación se denomina intruso. Este intruso puede ser pasivo, si sólo escucha la comunicación, o activo si trata de alterar los mensajes.

Es aquí donde aparece la criptografía con objeto de proporcionar comunicaciones seguras sobre canales inseguros. Los mensajes sin ninguna transformación se denominan “texto en claro”. El proceso mediante el cual la información contenida en el mensaje es ocultada se denomina encriptado. Un mensaje encriptado también se denomina “texto cifrado”. El proceso mediante el cual se revierte el proceso de ocultación, obteniéndose el texto en claro a partir del texto cifrado, se denomina desencriptado.

La criptografía trata de permitir que dos entidades, ya sean usuarios o aplicaciones, puedan enviarse mensajes por un canal que puede ser intervenido por una tercera entidad, de modo que sólo los destinatarios autorizados puedan leer los mensajes.

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, en el contexto de *Internet*, la confidencialidad es, a menudo, un factor secundario. Generalmente se estará más interesado en el mantenimiento de la integridad de los mensajes y en los mecanismos de autentificación, que implícitamente proporciona la criptografía. En efecto, un mensaje encriptado sólo puede ser desencriptado si la clave que se va a utilizar para ello pertenece a quien ha ocultado previamente el mensaje.

Criptosistema es una especie de quíntupla (M, C, K, E, D), donde: M (por “message”) representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto en claro, o *plaintext*) que pueden ser enviados son frecuentemente datos binarios. C (de “*ciphertext*”) representa el conjunto de todos los posibles mensajes cifrados, o

criptogramas. K representa el conjunto de claves que se pueden emplear en el criptosistema. E es el conjunto de transformaciones de cifrado, o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_k para cada valor posible de la clave k. D es el conjunto de transformaciones de descifrado, inverso de E.
Todo criptosistema ha de cumplir la siguiente condición:

$$D_k(E_k(m)) = m$$

es decir, que si se tiene un mensaje m, se cifra empleando la clave k y luego se descifra empleando la misma clave (cifrado simétrico), se obtiene de nuevo el mensaje original m.
Un algoritmo criptográfico es una función matemática utilizada para el cifrado y descifrado de mensajes. Generalmente, hay dos funciones relacionadas: una para el cifrado y otra para el descifrado.

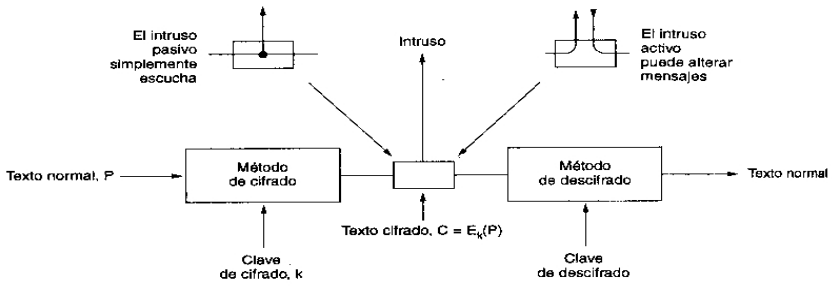


Figura 3-2. El modelo de cifrado

3.2.3 Criptografía de clave simétrica

Las técnicas de clave única, secreta o simétrica tienen fundamentos de complejidad diversa, pero todas usan una misma clave ***k*** que es conocida por el remitente de los mensajes y por el receptor, mediante la cual se encripta y desencripta el mensaje que se quiere proteger [38].

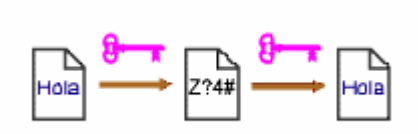


Figura 3-3. Criptografía de clave simétrica

Los cifradores simétricos pueden dividirse en dos grupos:

1. Cifradores de flujo, los cuales cifran un único bit del texto en claro cada vez.
2. Cifradores de bloque, que toman un grupo de bits y lo cifran como si se tratase de una unidad.

Una de las ventajas de la criptografía de clave simétrica es la existencia de algoritmos muy rápidos y eficientes, especialmente si se implementan en *hardware*. Si ***k*** es lo suficientemente larga (típicamente se usan valores de 56 a 256 bits), resulta casi imposible “romper” el sistema usando la llamada “fuerza bruta”.

El usar la misma clave para encriptar y para desencriptar es un problema a la hora de enviar datos, ya que el remitente debe enviar previamente la clave al destinatario para que éste pueda desencriptar la información, y debe hacerlo por un canal seguro, frecuentemente utilizando para este fin la criptografía asimétrica.

3.2.4 Criptografía de clave asimétrica

La solución al problema de la distribución de claves apareció en 1976 cuando Whitfield Diffie y Martín Hellman demostraron la posibilidad de construir sistemas criptográficos que no precisaban la transferencia de una clave secreta entre emisor y receptor, evitando así los problemas derivados de la búsqueda de canales seguros para la transferencia. Se trata de la "criptografía de clave asimétrica, o pública".

Se considera un esquema de encriptación como asimétrico, cuando cada conjunto, o pareja, de transformaciones de encriptado y desencriptado posee un par de claves (***e*** y ***d***) que únicamente sirve para una sola operación. En dicho esquema, una clave ***e*** es hecha pública, mientras la segunda se mantiene en secreto ***d***. Para que el esquema sea seguro, debe ser computacionalmente factible obtener ***d*** a partir de ***e***. [16]

Figura 3-4. Criptografía de clave asimétrica

Tanto la criptografía simétrica como la asimétrica basan su fortaleza en problemas matemáticos difíciles de resolver, como por ejemplo la factorización de números enteros grandes. Sin embargo, debido al avance en la potencia de computación, se estima en la actualidad que los sistemas asimétricos solamente ofrecen muy buena seguridad cuando las claves son de 2048 bits en el caso del algoritmo RSA.

3.2.5 Tipos de ataques

Una comunicación, ya sea protegida, o no, mediante sistemas criptográficos, está sujeta a una gran variedad de ataques, de los cuales son más habituales los siguientes [38]:

A) Ataque sólo 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 algoritmo de cifrado, sólo puede intentar vulnerar dicho algoritmo, realizar un análisis estadístico de los criptogramas, o probar todas las claves posibles del algoritmo. Este último caso, por motivos obvios se conoce como búsqueda exhaustiva, o también como ataque basado en la fuerza bruta.

B) Ataque mediante texto en claro conocido: en este ataque se tienen pares de texto en claro y su equivalente encriptado (muchos mensajes encriptados,

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.

C) Ataque mediante texto en claro escogido: el intruso es capaz de conseguir que un texto elegido por él sea cifrado con la clave desconocida. Por tanto, para efectos de protección, hay que diseñar el sistema criptográfico de modo que nunca un intruso pueda introducir mensajes propios.

D) Ataque adaptable mediante texto en claro escogido: el intruso no sólo puede elegir el texto que quiere cifrar, sino que puede tomar decisiones sobre el texto que será encriptado basándose en resultados anteriores.

E) Ataque mediante criptogramas escogidos: el atacante puede obtener el descifrado de diversos mensajes encriptados escogidos por él.

Por otra parte, en el marco de una comunicación entre dos entidades, se puede hablar de los siguientes ataques [16]:

a) Escucha pasiva ("*passive eavesdropping*"): el intruso simplemente escucha el tráfico que circula por el canal.

b) 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.

c) Retransmisión ciega ("*replay*"): el intruso intercepta un mensaje legítimo, lo almacena (sin eliminarlo) y lo reenvía un tiempo después.

d) 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 él.

e) Puesta a cero del reloj ("*time-resetting*"): en protocolos que utilizan de alguna forma la hora actual, el intruso puede tratar de confundir acerca de cuál es la verdadera hora.

Y en cuanto a los atacantes, la siguiente tabla resume los grupos de personas más comúnmente propensos a cometer ataques a sistemas informáticos y las razones por las que lo hacen.

ADVERSARIO	META
Estudiante	Divertirse husmeando el correo de la gente
Hacker	Probar el sistema de seguridad de alguien: robar datos
Representante de ventas	Hacerse pasar por alguien más
Hombre de negocios	Descubrir el plan estratégico de mercadeo de un competidor
Ex empleado	Vengar su despido

Contador	Estafar dinero de una compañía
Corredor de bolsa	Negar une promesa hecha a un cliente por correo electrónico
Timador	Robar números de tarjeta de crédito
Espía	Conocer la fuerza militar de un enemigo
Terrorista	Robar secretos de guerra

Tabla 3-2. Algunas personas que causan problemas de seguridad, y por qué [21].

3.2.6 Principios criptográficos fundamentales

Aunque analizaremos varios sistemas criptográficos diferentes en los siguientes puntos, se presentan dos principios que los sostienen a todos y que es importante tener en mente [21]:

1. Todos los mensajes deben contener redundancia para evitar que los intrusos activos engañen al receptor y lo hagan actuar ante un mensaje falso. Sin embargo, esta misma redundancia simplifica mucho la violación del sistema por parte de los intrusos pasivos, por lo que se recomienda usar una cadena aleatoria de palabras como mensaje de redundancia.

2. El segundo principio criptográfico es que deben tomarse algunas medidas para evitar que los intrusos activos reproduzcan mensajes viejos. Una de tales medidas es la inclusión en cada mensaje de una marca de tiempo válida durante un periodo.

3.3 Calidad de la Información y Virus Informáticos

Al momento de desarrollar un sitio de comercio electrónico se tienen que tomar en cuenta varios factores, que si bien los más importantes y que salen a relucir son los referentes a la seguridad en cuanto a las transacciones electrónicas, sin embargo no hay que olvidar que otros fundamentos también son importantes como son la calidad de la información, que nos permite tener un mejor desempeño a la hora de crear, desarrollar y actualizar el portal electrónico y también la protección contra los virus informáticos ya que no basta tener una comunicación segura si por otro lado es vulnerable a la proliferación de un virus.

3.3.1 Calidad de la información

Todo tipo de información es susceptible de ser evaluada, sobre todo si se requiere reunir una colección de utilidad para los usuarios de un centro de información. La información almacenada en los soportes tradicionales, e incluso en los electrónicos, cuenta desde hace tiempo con un conjunto teórico, relativo a los criterios que se deben aplicar para la evaluación de la misma. Sin embargo, la información telemática, especialmente la accesible a través de *Internet*, todavía está siendo objeto de reflexión e investigación, a fin

de ofrecer una serie de parámetros y procedimientos que sirvan de forma definitiva para analizar la calidad de la información accesible en línea [33].

La información en el comercio electrónico es aquella que está elaborada en cualquiera de los lenguajes derivados del HTML y cuya característica más notable consiste en ser documentos hipertextuales y multimedia, indispensables en el desarrollo de un sito de comercio electrónico, ya que de acuerdo a la información recaudada se pueden plantear estrategias para un mejor desempeño en la elaboración del portal electrónico, tal y como se menciono en el capitulo anterior. La unidad básica de los documentos de este tipo es la página *Web*, entendida como el documento escrito en un lenguaje de marcado, con una localización única dentro de un servidor. El contenido de una página *Web* puede ser independiente o bien estar vinculado a otras páginas *Web*, entre las que existen enlaces hipertextuales y las cuales completan su información. En este caso, se denomina sitio *Web*, al conjunto de páginas encargadas de ofrecer la venta de algún producto (por ejemplo: música) de manera encriptada. Así como para conocer el gusto de los usuarios para determinado genero de música y poder contar con un sitio más interesante para los posibles compradores. Esta delimitación de conceptos es importante, ya que el proceso de evaluación de información telemática muchas veces podrá realizarse sobre páginas aisladas aunque, en la mayoría de los casos, el objeto será un sitio *Web* en su conjunto y con mayor calidad.

La evaluación de páginas o sitios *Web* es necesaria por motivos cuantitativos y cualitativos [33].

Asimismo, cualquier fuente de información sólo es válida si aporta contenidos útiles y si los mismos son localizados de forma sencilla. Por este motivo, también es necesario recurrir a parámetros que ayuden a identificar la información imprescindible y separarla de la que no aporta gran cosa. Es evidente que es sin lugar a dudas necesario disponer de indicadores para aplicar en el proceso de evaluación.

3.3.2 Proceso de evaluación de la información

La evaluación de la información telemática, como la de cualquier otro tipo, requiere una planificación concreta en la que se establecerán los criterios que se aplicarán y los métodos mediante los que se pondrán en práctica dichos criterios. Los criterios se materializarán mediante el uso de parámetros e indicadores de evaluación; mientras que lo métodos se desarrollan a través de procedimientos concretos y la ayuda de los recursos necesarios para la realización positiva de los métodos ideados para llevar a cabo el proceso de evaluación. Parámetros, indicadores, procedimientos y recursos son, por tanto, los cuatro elementos clave del proceso de evaluación de la información *Web*.

Los *parámetros* son los aspectos genéricos que serán evaluados. Se trata de establecer una serie de grandes bloques sobre los que se realizará el análisis y los cuales serán desarrollados por indicadores concretos que dan la información necesaria para cada uno de estos grupos.

Los *indicadores* son los elementos que desarrollan cada uno de los parámetros establecidos para el análisis de la información. Son las cuestiones concretas que se

evaluarán. Como ocurre con los parámetros, existen múltiples componentes que pueden ser considerados como un índice de la calidad de una página, o de un sitio *Web*.

Los *procedimientos* son los métodos que se emplean para hacer efectiva la aplicación de parámetros e indicadores. Este es el aspecto del proceso de evaluación que presenta un menor grado de desarrollo en cuanto a aportaciones teóricas o experiencias prácticas, ya que sólo hay propuestas aisladas y parciales. La planificación de cualquier proceso de evaluación no puede limitarse a delimitar qué se debe analizar, sino también debe decir cómo se debe obtener la información relativa a los elementos que se están evaluando.

Los *recursos* son los materiales necesarios para el proceso de evaluación. Conocidos qué aspectos serán evaluados y cómo se procederá a su análisis, será necesario establecer qué medios humanos, instrumentales y documentales son necesarios. Como ocurría con los procedimientos, los recursos también están poco estructurados y, por lo general, en la planificación y ejecución de la evaluación, sólo se contemplan los recursos humanos y algunos documentales como las listas de parámetros, e indicadores; los formularios o plantillas de análisis.

3.3.3 Virus informáticos

Un virus es un programa diseñado para dañar sistemas informáticos, alterando su forma de trabajar, o dañando información almacenada en el disco duro. Por supuesto, sin el conocimiento, o permiso, del afectado.

En términos más técnicos, un virus se define como una porción de código de programación cuyo objetivo es convertirse a si mismo en un archivo ejecutable y multiplicarse sistemáticamente de un archivo a otro. Además de esta función primaria de "invasión" o "reproducción", los virus están diseñados para realizar una acción concreta en los sistemas informáticos. Esta acción puede ir desde la simple aparición de un mensaje en la pantalla, hasta la destrucción de toda la información contenida en el sistema.

El ciclo de los virus informáticos es muy similar al de los biológicos.

- *Infección:* Al ejecutar un archivo infectado (el código del virus se ha implantado en el archivo anteriormente) comienza la fase de infección, duplicándose e implantándose en otros archivos ejecutables. Se pasa a la fase de "invasión" del sistema informático. La víctima, aún no es consciente de la existencia del virus ya que este permanece oculto y sin causar daños apreciables.

- *Expansión:* El virus pasará a otras computadoras, a través de redes informáticas, *disquetes* y CDs que contengan archivos infectados, *software* en *Internet*, archivos adjuntos a mensajes electrónicos, etc.

- *Explosión:* Si el virus no ha sido detectado y destruido por algún *programa antivirus*, en un momento determinado, o bajo determinadas circunstancias, tomará el control de la computadora infectada, ejecutando la acción para la que fue programado. En ese momento, debido a los trágicos efectos que puede llegar

a ocasionar, se hará evidente su existencia, acabando con información vital contenida en el sistema informático.

3.3.4 Clasificación de virus

Dentro del término "virus informático" se suelen englobar varios tipos de programas, por lo que a continuación se da un pequeño repaso a cada uno de ellos, poniendo de manifiesto sus diferencias. La clasificación es la siguiente:

Virus Puro. Un verdadero virus tiene como características más importantes la capacidad de copiarse a sí mismo en soportes diferentes al que se encontraba originalmente, y por supuesto hacerlo con el mayor sigilo posible y de forma transparente al usuario. Como soporte se entiende el lugar donde el virus se oculta, ya sea archivo, sector de arranque, partición, etc. Un virus puro también debe modificar el código original del programa, o soporte objeto de la infección, para poder activarse durante la ejecución de dicho código; al mismo tiempo, una vez activado, el virus suele quedar residente en memoria para poder infectar así de forma transparente al usuario.

Caballo de Troya. Al contrario que el virus puro, el llamado “Caballo de Troya” es un programa maligno que se oculta en otro programa legítimo, y que produce sus efectos perniciosos al ejecutarse este último. En este caso, no es capaz de infectar otros archivos, o soportes, y sólo se ejecuta una vez, aunque es suficiente, en la mayoría de las ocasiones, para causar su efecto destructivo.

Bomba Lógica. Se trata simplemente de un programa maligno que permanece oculto en memoria y que solo se activa cuando se produce una acción concreta, predeterminada por su creador: cuando se llega a una fecha en concreto, cuando se ejecuta cierto programa, o cierta combinación de teclas, etc.

Gusano o Word. Por último, un gusano en un programa cuya única finalidad es la de ir consumiendo la memoria del sistema, mediante la realización de copias sucesivas de sí mismo, hasta desbordar la RAM, siendo ésta su única acción maligna.

La barrera entre virus puros y el resto de programas malignos es muy difusa, prácticamente invisible, puesto que ya casi todos los virus incorporan características propias de uno o de varios de estos programas.

3.3.5 Prevención, detección y eliminación de virus

Una buena política de prevención y detección es necesaria. Las medidas de prevención pasan por el control, en todo momento, del *software* ya introducido, o que se va a introducir en nuestra computadora, comprobando la fiabilidad de su fuente. Esto implica el escaneo, con un buen programa antivirus, de todo el *software* que llega, y ante la más mínima duda lo mejor es deshacerse inmediatamente de este.

Por supuesto, el sistema operativo, que a fin de cuentas es el elemento de *software* más importante del ordenador, debe ser totalmente fiable; si éste se encuentra infectado, cualquier programa que se ejecute resultara también contaminado. Por eso, es

imprescindible contar con una copia protegida del sistema operativo; esto último es muy importante, no solo con el sistema operativo, sino con el resto de los programas que se posean. Por último es también imprescindible poseer un buen *software* antivirus con su respectiva actualización, que detecte y elimine cualquier tipo de intrusión en el sistema.

3.4 Introducción a la Seguridad Física

Las primeras medidas de seguridad que se necesita tener en cuenta son las referentes a la seguridad física de los sistemas. Hay que tomar en consideración a quiénes tienen acceso físico a las máquinas y si realmente deberían tenerlo.

La seguridad física de los sistemas informáticos consiste en la aplicación de barreras físicas y procedimientos de control como medidas de prevención y contramedidas contra las amenazas a los recursos y la información confidencial. Más claramente, y particularizando para el caso de equipos Servidores y sus centros de operación, por seguridad física se puede entender todos aquellos mecanismos - generalmente de prevención y detección - destinados a proteger físicamente cualquier recurso del sistema; estos recursos son desde un simple teclado hasta una cinta de *backup* con toda la información que hay en el sistema, pasando por la propia CPU de la máquina [44].

Desgraciadamente, la seguridad física es un aspecto olvidado con demasiada frecuencia a la hora de hablar de seguridad informática en general; en muchas organizaciones se suelen tomar medidas para prevenir o detectar accesos no autorizados o negaciones de servicio, pero rara vez para prevenir la acción de un atacante que intente acceder físicamente a la sala de operaciones, o al lugar donde se depositan las impresiones del sistema. Esto motiva que en determinadas situaciones un atacante se incline por aprovechar vulnerabilidades físicas en lugar de lógicas, ya que posiblemente le sea más fácil robar una cinta con una imagen completa del sistema, que intentar acceder a él mediante fallas en el *software*. La seguridad física es demasiado importante como para ignorarla: un ladrón que roba una computadora para venderla, un pirata que accede sin problemas a la sala de operaciones pueden hacer mucho más daño que un intruso que intenta conectarse remotamente con una máquina no autorizada; no importa que se utilicen los más avanzados medios de cifrado para conectar a los servidores, ni que se haya definido una política de *firewall* muy restrictiva: si no se tiene en cuenta factores físicos, estos esfuerzos para proteger la información no van a servir de mucho.

La posibilidad de acceder físicamente a una máquina hace inútiles casi todas las medidas de seguridad que se hayan aplicado sobre ella: si un atacante puede llegar con total libertad hasta una estación puede por ejemplo abrir la CPU y llevarse un disco duro; sin necesidad de privilegios en el sistema, sin importar la robustez del cortafuegos (*firewall*), sin ni siquiera una clave de usuario, el atacante podrá seguramente modificar la información almacenada, destruirla o simplemente leerla. Incluso sin llegar al extremo de desmontar la máquina, que quizás resulte algo difícil en entornos clásicos donde hay cierta vigilancia, como un laboratorio o una sala de informática, la persona que accede al equipo puede detenerlo, o arrancar una versión diferente del sistema operativo sin llamar mucho la atención.

3.4.1 Prevención y detección de intrusos

Hay diversas soluciones para prevenir y detectar la entrada de intrusos, son de diversa índole, y de diferentes precios: desde analizadores de retina hasta videocámaras, pasando por tarjetas inteligentes, o control de las claves que abren determinada puerta. Todos los modelos de autenticación de usuarios son aplicables, aparte de para controlar el acceso lógico a los sistemas, también para controlar el acceso físico de todos ellos, quizás los más adecuados a la seguridad física sean los biométricos; aunque suelen resultar algo caros para utilizarlos masivamente en entornos de seguridad media.

Pero sin una razón válida, no conviene orientarse a sistemas tan complejos para prevenir accesos físicos no autorizados; normas tan elementales como cerrar las puertas con clave al salir de un laboratorio, o un despacho, o bloquear las conexiones de red que no suelen utilizarse y que estén situadas en lugares apartados, son en ocasiones más que suficientes para prevenir ataques. También basta el sentido común para darse cuenta de que el cableado de red es un elemento importante para la seguridad, por lo que es recomendable apartarlo del acceso directo.

Cuando la prevención es difícil por cualquier motivo (técnico, económico, humano...) es deseable que un potencial ataque sea detectado cuanto antes, para minimizar así sus efectos. Aunque en la detección de problemas, generalmente accesos físicos no autorizados, intervienen medios técnicos, como cámaras de vigilancia de circuito cerrado, o alarmas, en entornos más normales el esfuerzo en detectar estas amenazas se ha de centrar en las personas que utilizan los sistemas y en las que sin utilizarlos están relacionadas de cierta forma con ellos; sucede lo mismo que con la seguridad lógica: se ha de ver toda la protección como una cadena donde cualquier eslabón es susceptible de ser atacado.

Es importante concienciar a todos de su papel en la política de seguridad del entorno; si por ejemplo un usuario autorizado detecta presencia de alguien de quien sospecha que no tiene autorización para estar en una determinada estancia, debe avisar inmediatamente al administrador de los equipos, quien a su vez puede avisar al servicio de seguridad si es necesario. No obstante, utilizar este servicio debe ser solamente un último recurso: generalmente en la mayoría de entornos no se esta tratando con terroristas, sino por fortuna con elementos mucho menos peligrosos.

3.4.2 Desastres: naturales y de entorno

En el punto anterior se ha hecho referencia a accesos físicos no autorizados a zonas, o a elementos que pueden comprometer la seguridad de los equipos, o de toda la red; sin embargo, no son estas las únicas amenazas relacionadas con la seguridad física. Un problema que no suele ser tan habitual, pero que en caso de producirse puede acarrear gravísimas consecuencias, es el derivado de los desastres naturales y su falta de prevención. Entre los desastres naturales se pueden destacar los siguientes:

- Terremotos
- Tormentas eléctricas
- Inundaciones y humedad

Por otro lado los problemas derivados del entorno pueden ser el reflejo de una mala instalación del equipo tanto de computo, como eléctrico, o de una mala ubicación de estos; lo cual puede acarrear a la pérdida del equipo y de la información, y es por esta razón que se tiene que poner mucha atención, ya que estas complicaciones son viables de poder ser prevenidas, sin la necesidad de excesivos gastos. A continuación se mencionan los principales desastres que se pueden presentar en un cuarto de telecomunicaciones:

- Eléctricos
- Ruido eléctrico
- Incendios y humos
- Temperaturas extremas

3.5 Criptografía de Clave Simétrica o de Clave Privada

3.5.1 *El algoritmo* DES

En enero de 1977, el gobierno de Estados Unidos adoptó un cifrador de bloque desarrollado por IBM como su estándar oficial para información no clasificada. Este cifrado, es conocido con el nombre de DES (*Data Encryption Standard*, estándar de cifrado de datos), se adoptó ampliamente en la industria para usarse con productos de seguridad. Ya no es seguro en su forma original, pero aún es útil en una forma modificada.

En la figura 3-2 se muestra un esbozo de este algoritmo. El texto en claro se ordena en bloques de 64 bits, produciendo 64 bits de texto cifrado. El algoritmo, utiliza una clave de 56 bits, tiene 16 etapas diferentes, llamadas rondas. Al inicio se hace una permutación, independiente de la clave, del texto normal de 64 bits. Y al final se hace una permutación inversa. En la última ronda no se intercambian los 32 bits de la izquierda y los 32 bits de la derecha del bloque a procesar. Salvo esta excepción, todas las 16 rondas son funcionalmente idénticas, pero se trabajan con diferentes valores de la clave (subclaves). El algoritmo se ha diseñado para permitir que el descifrado se haga con la misma clave que el cifrado, de tal manera que es simétrico. Los pasos simplemente se ejecutan en orden inverso.

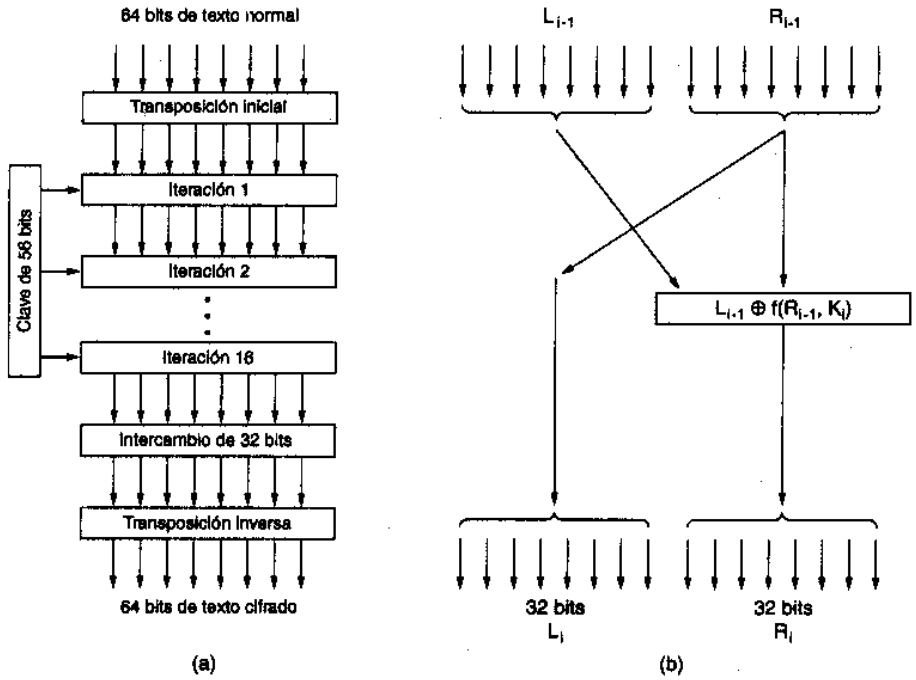


Figura 3-2. DES (a) Esbozo general (b) Detalle de una ronda

La operación de una de estas rondas intermedias se ilustra en la figura 3-2(b). Cada etapa toma dos entradas de 32 bits y produce dos salidas de 32 bits. En cada ronda, la salida de la izquierda simplemente es una copia de la entrada de la derecha. La salida de la derecha es el OR EXCLUSIVO a nivel de bit de la entrada izquierda y una función de la entrada derecha y de la clave de esta etapa, K_i. Toda la complejidad reside en esta función.

3.5.1.1 Encadenamiento DES y modos de operación

A pesar de toda esta complejidad, el DES básicamente es un cifrado por sustitución monoalfabética que usa un carácter de 64 bits. Cada vez que entra el mismo bloque de texto normal de 64 bits, sale el mismo bloque de texto cifrado de 64 bits. Un criptoanalista puede explotar esta propiedad como ayuda para violar el DES.

Para ver la manera en que esta propiedad de cifrado por sustitución monoalfabética puede usarse para subvertir al DES, se debe considerar el cifrado de un mensaje grande de la manera obvia: dividiéndolo en bloques consecutivos de 8 bytes (64 bits) y cifrándolos uno tras otro con la misma clave. El último bloque se rellena hasta completar los 64 bits, de ser necesario. Esta técnica se conoce como modo de libro de código electrónico.

En la figura 3-3 se puede observar el comienzo de un archivo de computadora que lista los bonos anuales que ha decidido otorgar una compañía a sus empleados. Este archivo consiste en registros consecutivos de 32 bytes, uno por empleado. En el formato que se muestra se tienen: 16 bytes para el nombre, 8 bytes para el puesto y 8 bytes para el bono. Cada uno de los 16 bloques (numerados del O al 15) de 8 bytes (64 bits) se cifra con el DES.

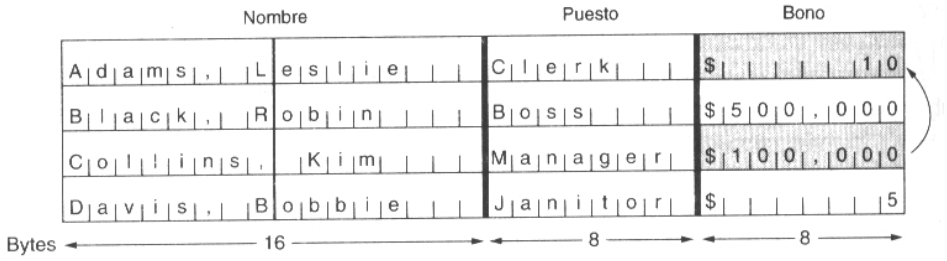


Figura 3-3. El texto normal de un archivo cifrado como 16 bloques DES [21].

Como puede verse, dado que el cifrado de la información se efectúa por bloques, un empleado malicioso puede, en un momento dado, tomar únicamente un bloque de información del archivo de la nomina y sustituirlo por otro. Con lo cual, estaría efectuando un fraude electrónico.

Para frustrar este tipo de ataque, el DES (y todos los cifradores de bloque) puede encadenarse de varias maneras para que el reemplazo de un bloque haga que el texto normal descifrado comenzando por el bloque reemplazado sea basura. Una forma de encadenar es por encadenamiento de bloque cifrado. En este modo, que se muestra en la figura 3-4, a cada bloque de texto normal se le hace un OR EXCLUSIVO (#) con el bloque de texto cifrado previo antes de cifrarse. En consecuencia, el mismo bloque de texto normal no corresponde con el mismo bloque de texto cifrado, y el cifrado ya no es un cifrado por sustitución monoalfabética grande. Al primer bloque se le hace un OR EXCLUSIVO con un **vector de inicialización, IV**, seleccionado al azar, que se transmite junto con el texto cifrado.

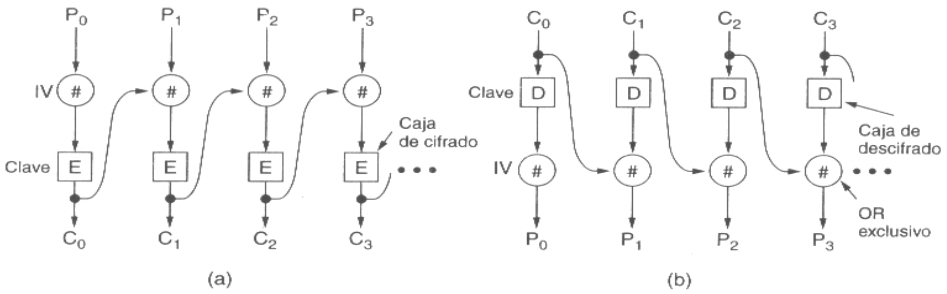


Figura 3-4. Encadenamiento de bloques cifrados [21]

Se puede observar cómo funciona el encadenamiento de bloques cifrados examinando el ejemplo de la figura 3-4. Comenzando por calcular $C_0 = E(P_0 \text{ XOR } IV)$. Después calculando $C_1 = E(P_1 \text{ XOR } C_0)$. etc. El descifrado funciona de la manera opuesta, con $P_0 = IV \text{ XOR } D(C_0)$, etc. Nótese que el cifrado del bloque *i* es una función de todo el texto normal de los bloques del 0 al *i - 1*, por lo que el texto normal genera un texto cifrado diferente dependiendo de dónde ocurre.

El encadenamiento de bloques cifrados también tiene la ventaja de que el mismo bloque de texto normal no produce el mismo bloque de texto cifrado, dificultando el criptoanálisis. De hecho, ésta es la razón principal de su uso.

Sin embargo, el encadenamiento de bloques cifrados tiene la desventaja de requerir la llegada de un bloque completo de 64 bits antes de poder iniciar el descifrado. Para el

cifrado byte por byte puede usarse el modo de realimentación de cifrado, que se ilustra en la figura 3-5. Al llegar el byte de texto normal 10, como se aprecia en la figura 3-5(a), el algoritmo DES opera con el registro de desplazamiento de 64 bits para generar un texto cifrado de 64 bits. Se extrae el byte de la izquierda de ese texto cifrado y se le aplica un OR EXCLUSIVO con P_{10}. Ese byte se envía por la línea de transmisión. Además, el registro de desplazamiento se desplaza a la izquierda 8 bits, causando la expulsión de C_2 por la izquierda y la introducción de C_{10} en la posición que C_9 dejó vacante del lado derecho. Nótese que el contenido del registro de desplazamiento depende de la historia previa completa del texto normal, por lo que un patrón que se repite varias veces en el texto normal se cifrará de manera diferente en cada ocasión. Como ocurre con el encadenamiento de bloques cifrados, se requiere un vector de inicialización para echar a andar el mecanismo.

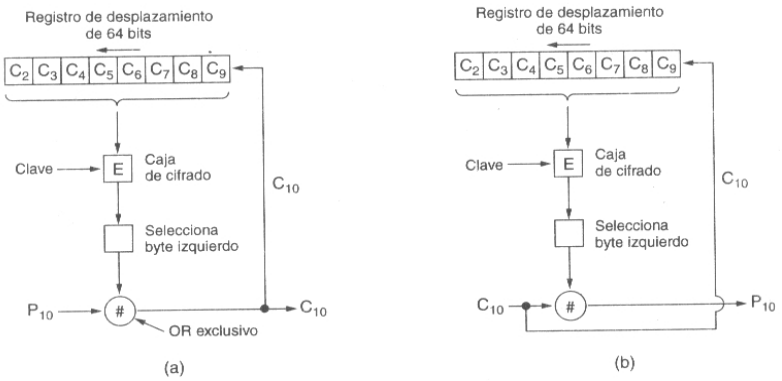


Figura 3-5. Modo de realimentación de cifrado [21]

El desencriptado con modo de realimentación de encriptado simplemente hace lo mismo que el encriptado. Mientras los dos registros de desplazamiento permanezcan idénticos, el descifrado funcionará correctamente.

Como nota al margen, debe indicarse que, si un bit del texto cifrado accidentalmente se invierte durante la transmisión, los 8 bytes descifrados cuando el byte malo esté en el registro de desplazamiento tendrán error. Una vez que el byte equivocado sea expulsado del registro de desplazamiento, se generará nuevamente texto normal correcto. Por tanto, el efecto de un solo bit invertido está delimitado a 8 bytes y no arruina el resto del mensaje.

Sin embargo, existen aplicaciones en las que un error de transmisión de 1 bit que arruina 64 bits de texto normal es un efecto demasiado grande. Para estas aplicaciones existe una cuarta opción, el modo de realimentación de salida, que es idéntico al modo de realimentación de cifrado, excepto que el *byte* realimentado por el lado derecho del registro de desplazamiento se toma justo antes de la caja de OR EXCLUSIVO, no justo después.

El modo de realimentación de salida tiene la propiedad de que un error de 1 bit en el texto cifrado causa un error de 1 solo bit en el texto normal resultante. Por otra parte, es menos seguro que los otros modos, y debe evitarse su uso de propósito general. El modo de libro de código electrónico también debe evitarse excepto en circunstancias especiales (por ejemplo, el cifrado de un solo número aleatorio, como una clave de sesión). Para la operación normal, debe usarse el encadenamiento de bloques cifrados cuando la entrada

llega en unidades de 8 bytes (por ejemplo, para cifrar archivos de disco) y el modo de realimentación de cifrado debe usarse para cadenas de entrada irregulares, como la entrada de un teclado.

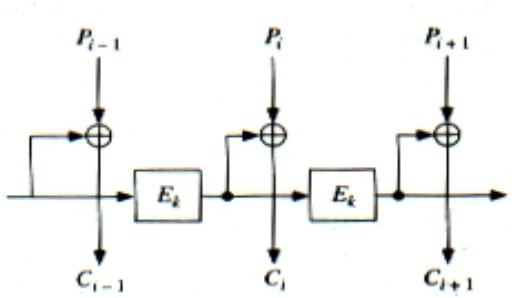


Figura 3-6. Modo de realimentación de salida [20]

3.5.1.2 Descifrado del DES

El DES ha estado rodeado de controversias desde el día en que se propuso; se basó en un cifrado desarrollado y patentado por IBM, llamado Lucifer, excepto que el cifrado de IBM usaba una clave de 128 bits en lugar de 56 bits. Cuando el gobierno de Estados Unidos quiso estandarizar un método de cifrado para uso no confidencial, "invitó" a IBM a "debatir" el asunto con la NSA (*"National Security Agency"*. Agencia Nacional de Seguridad).

Tras estos debates, la IBM redujo la clave de 128 a 56 bits y decidió mantener en secreto el proceso de diseño del DES. Mucha gente sospechó que la longitud de la clave se redujo para asegurar que la NSA pudiera descifrar el código, pero no así alguna otra organización con menor presupuesto. El objetivo del diseño secreto supuestamente era esconder una puerta secreta que pudiera facilitar aún más el descifrado del DES por la NSA [21].

DES puede ser atacado mediante la fuerza bruta, probando todas las claves posibles (2^{56}), siendo este algoritmo de una complejidad $O(2^{55})$. A pesar de los rumores que aseguraban que el NSA modificó el algoritmo para hacerlo más débil, aún no ha sido roto públicamente más que por la fuerza bruta.

Probablemente la idea más innovadora para descifrar el DES es la lotería china (conceptualizada por Quisquater y Girault en 1991). En este diseño, cada radio y televisión tiene que equiparse con un chip DES barato capaz de realizar 1 millón de cifrados por segundo en "*hardware*". Suponiendo que cada una de las 1.2 mil millones de personas de China tiene un radio o televisión, cada vez que el gobierno quiera descifrar un mensaje codificado con DES, simplemente difunde el par texto normal / texto cifrado, y cada uno de los 1.2 mil millones de chips comienzan a buscar su sección preasignada del espacio de claves. En 60 segundos se encontrarán una o más correspondencias. Para asegurar que se informen, los chips podrían programarse para desplegar o anunciar el mensaje: ¡FELICIDADES! ACABA DE GANAR LA LOTERÍA CHINA. PARA COBRAR EL PREMIO, POR FAVOR LLAME AL 1-800-GRAN-PREMIO.

La conclusión que se puede obtener de estos argumentos es que el DES ya no debería usarse para nada importante. Sin embargo, aunque 2^{56} es 7×10^{16}, 2^{112} es 5×10^{33}. Aun con mil millones de chips DES efectuando mil millones de operaciones por segundo, se

requerirían 100 millones de años para examinar detalladamente un espacio de claves de 112 bits. Por tanto, surge la idea de simplemente ejecutar el DES dos veces, con dos claves de 56 bits diferentes.

Desgraciadamente, Merkle y Hellman (1981) han desarrollado un método que hace sospechoso al doble cifrado. Se llama ataque de encuentro a la mitad ("meet-in-the-middle") y funciona como sigue (Hellman. 1980). Supóngase que alguien ha cifrado doblemente una serie de bloques de texto normal usando el modo de libro de código electrónico. Para unos pocos valores de *i* el criptoanalista tiene pares igualados (P_i, C_i) donde

$$C_i = E_{K2}(E_{K1}(P_i))$$

Si ahora se aplica la función de descifrado, D_{K2}, a cada lado de esta ecuación, se obtiene:

$$D_{K2}(C_i) = E_{K1}(P_i) \qquad (I)$$

dado que el cifrado de ***x*** y su descifrado posterior con la misma clave produce ***x***.

El ataque de encuentro a la mitad usa esta ecuación para encontrar las claves DES, *K1* y *K2*, como sigue:

1. Calcular $R_i = E_i(P_1)$ para los 2^{56} valores de *i*, donde *E* es la función de cifrado DES. Ordenar esta tabla en orden ascendente según R_i.

2. Calcular $S_j = D_j(C_1)$ para todos los 2^{56} valores de *j*, donde *D* es la función de descifrado DES. Ordenar esta tabla en orden ascendente según S_j.

3. Barrer la primera tabla en busca de un R_i igual a algún S_j de la segunda tabla. Al encontrar un par, se tiene un par de claves (*i, j*) tal que $D_j(C_1) = E_i(P_1)$. Potencialmente, *i* es *K1* y *j* es *K2*.

4. Comprobar si $E_j (E_i (P_2))$ es igual a C_2. Si lo es, intentar todos los demás pares (texto normal, texto cifrado). De no serlo, continuar buscando pares en las dos tablas.

Ciertamente ocurrirán muchas falsas alarmas antes de encontrar las claves reales, pero tarde o temprano se encontrarán. Este ataque requiere sólo 2^{57} operaciones de cifrado o descifrado (para construir las dos tablas), mucho menos que 2^{112}; sin embargo, también requiere un total de 2^{60} bytes de almacenamiento para las dos tablas, por lo que actualmente no es factible en su forma básica, pero Merkle y Hellman han mostrado varias optimizaciones y concesiones que permiten menos almacenamiento a expensas de más cómputo. En conclusión, el cifrado doble usando el DES probablemente no es mucho más seguro que el cifrado sencillo.

3.5.1.3 DES múltiple

Para 1979, IBM se dio cuenta de que la longitud de la clave DES era demasiado corta y diseñó una manera de aumentarla efectivamente usando codificación múltiple (modificación ideada por Tuchman). El más común de todos ellos es el Triple-DES, que responde a la siguiente estructura:

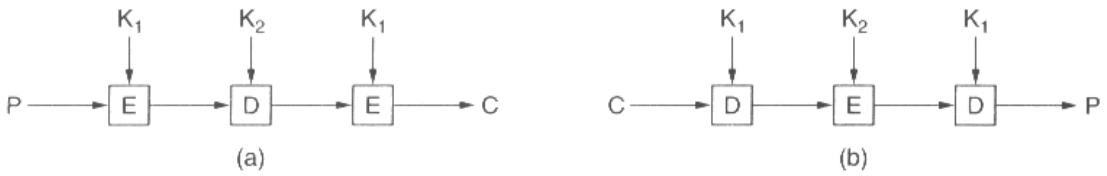


Figura 3-7. Cifrado triple usando el DES. [21]

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

es decir, se codifica con la subclave *k1*, se decodifica con *k2* y se vuelve a codificar con *k1*. La clave resultante es la concatenación de *k1* y *k2*, con una longitud de 112 bits.

Este diseño inmediatamente da pie a dos preguntas. Primero, ¿por qué sólo se usan dos claves en lugar de tres? Segundo, ¿por qué se usa EDE (cifrado-descifrado-cifrado) en lugar de EEE (cifrado-cifrado-cifrado)? La razón de que se usen dos claves es que incluso los criptógrafos más paranoicos coinciden en que 112 bits son suficientes para las aplicaciones comerciales por ahora. Subir a 168 bits simplemente agregaría la carga extra innecesaria de administrar y transportar otra clave.

La razón para cifrar, descifrar y luego cifrar de nuevo es la compatibilidad en reversa con los sistemas DES de una sola clave. Tanto las funciones de cifrado como de descifrado son correspondencias entre grupos de números de 64 bits. Desde el punto de vista criptográfico, las dos correspondencias son igualmente robustas. Sin embargo, usando EDE en lugar de EEE, una computadora que usa cifrado triple puede entenderse con otra que usa cifrado sencillo simplemente estableciendo *K1 = K2*. Esta propiedad permite la introducción gradual del cifrado triple, algo que no interesa a los criptógrafos académicos, pero de importancia considerable para IBM y sus clientes.

No se conoce ningún método para descifrar el DES triple en modo EDE. Van Oorschot y Wiener (1988) han presentado un método para acelerar la búsqueda de EDE en un factor de 16, pero aun con su ataque, el EDE es muy seguro. Para alguien que desea algo mejor, se recomienda el EEE con tres diferentes claves de 56 bits (168 bits en total).

Antes de dejar el tema del DES, vale la pena cuando menos mencionar dos recientes avances del criptoanálisis. El primero es el criptoanálisis diferencial (dado a conocer por Biham y Shamir en 1993). Esta técnica puede usarse para atacar cualquier cifrado en bloques; funciona comenzando por un par de bloques de texto normal que difieren sólo en una cantidad pequeña de bits y observando cuidadosamente lo que ocurre en cada iteración interna a medida que avanza el cifrado. En muchos casos, algunos patrones son mucho más comunes que otros, y esta observación conduce a un ataque probabilístico.

El otro avance que vale la pena mencionar es el criptoanálisis lineal (ideado por Matsui en 1994). Que puede descifrar el DES con sólo 2^{43} textos comunes conocidos. Funciona haciendo un OR EXCLUSIVO entre ciertos bits de texto normal y texto cifrado para generar 1 bit. Al hacerse repetidamente, la mitad de los bits deben ser ceros y la otra deben ser unos. Sin embargo, con frecuencia los cifrados introducen un sesgo en una dirección o en la otra, y este sesgo, por pequeño que sea, puede explotarse para reducir el factor de trabajo.

Existen varias implementaciones de triple DES:

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

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

3.5.2 Algoritmo Rijndael (AES)

En octubre de 2000 el *National Institute for Standards and Technology* (NIST) anunció oficialmente la adopción del algoritmo Rijndael como nuevo Estándar Avanzado de Cifrado (AES) para su empleo en aplicaciones criptográficas no militares, culminando así un proceso de mas de tres años, encaminado a proporcionar a la comunidad internacional un nuevo algoritmo de cifrado potente, eficiente, y fácil de implementar, para reemplazar al DES.

El algoritmo AES es un sistema de cifrado por bloques, diseñado para manejar longitudes de clave y de bloque variables, estas pueden tomar valores de 128, 192 y 256 bits. Realiza varias de sus operaciones internas a nivel de byte, interpretando éstos como elementos de un campo de Galois GF(2^8). El resto de operaciones se efectúan en términos de registros de 32 bits. Sin embargo, en algunos cálculos, una secuencia de 32 bits se toma como un polinomio de grado inferior a 4, cuyos coeficientes son elementos de GF (2^8).

Si bien, como ya se ha dicho, este algoritmo soporta diferentes tamaños de bloque y clave, en el estándar adoptado por el gobierno Estadounidense en noviembre de 2001 (FIPS PUB 197), se especifica una longitud de clave fija de 128 bits.

3.5.2.1 Estructura de AES

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

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

- La capa no lineal —función ByteSub—consiste en la aplicación paralela de s-cajas con propiedades óptimas de no linealidad.

- La capa de adición de clave es un simple or-exclusivo entre el estado intermedio y la subclave correspondiente a cada ronda.

AES es un algoritmo que se basa en aplicar un número determinado de rondas a un cálculo progresivo que se denomina estado. Dicho estado puede representarse mediante una matriz rectangular de bytes, que posee cuatro filas, y N_b columnas. Así, por ejemplo, si el bloque tiene 160 bits ver tabla 3.3, N_b será igual a 5.

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

Tabla 3-3 Ejemplo de matriz de estado con N_b=5 (160 bits).

La clave tiene una estructura similar a la del estado, y se representara mediante una tabla con cuatro filas y N_k columnas. Si la clave tiene, por ejemplo, 128 bits, N_k será igual a 4.

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

Tabla 3-4 Ejemplo de clave con N_k=4 (128 bits).

En algunos casos, tanto el estado como la clave se consideran como vectores de registros de 32 bits, estando cada registro constituido por los 4 bytes de la columna correspondiente, ordenados de arriba a abajo.

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

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

1. Calcular K_0, K_1,….. K_n subclaves a partir de la clave K.
2. $S \leftarrow B \oplus K_0$
3. Para i = 1 hasta n, hacer el correspondiente numero de rondas, (ver tabla)

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

Tabla 3-5 Número de rondas para AES en función de los tamaños de clave y bloque.

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

Puesto que cada ronda es una sucesión de funciones invertibles, el algoritmo de descifrado consistirá en aplicar las funciones inversas de cada una de las funciones en el orden contrario, y utilizar las mismas subclaves K_i que en el cifrado, sólo que en orden inverso.

AES permite emplear diferentes longitudes tanto de bloque como de clave, el número de rondas requerido en cada caso es variable. En la tabla 3.5 se especifica cuántas rondas son necesarias en función de los valores de N_b y N_k.

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

Tabla 3-5 Número de rondas para AES en función de los tamaños de clave y bloque.

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

1. S ← ByteSub(S)
2. S ← DesplazarFila(S)
3. S ← MezclarColumnas(S)
4. S ← $K_i \oplus$S

La última ronda es igual a las anteriores, pero no usa el paso 3.

Función ByteSub

La transformación ByteSub es una sustitución no lineal que se aplica a cada byte de la matriz de estado, mediante una de las llamadas caja-s 8*8 invertible, que se obtiene aplicando las transformaciones siguientes:

1. Cada byte es considerado como un elemento del campo de Galois GF (2^8) (el cual es generado por el polinomio irreducible $m(x) = x^8 + x^4 + x^3 + x + 1$), y es sustituido por su inverso multiplicativo. Para evitar una indeterminación, el valor cero queda inalterado.

2. Después se aplica la siguiente transformación *afín* definida sobre el campo de Galois de orden 2, GF (2), Siendo $x_0, x_1, \ldots, x_7$ los bits del byte correspondiente, e $y_0, y_1, \ldots, y_7$ los del resultado:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

La función inversa de ByteSub se calcularía aplicando la inversa de la caja-s correspondiente a cada byte de la matriz de estado.

Función DesplazarFila

Esta transformación consiste en desplazar a la izquierda cíclicamente las filas de la matriz de estado. Cada fila f_i se desplaza un número de posiciones C_i diferente. Mientras que c_0 siempre es igual a cero (esta fila siempre permanece inalterada), el resto de valores está en función de N_b y se proporciona en la tabla 3.6

N_b	C1	C2	C3
4	1	2	3
6	1	2	3
8	1	3	4

Tabla 3-6 Valores de C_i según el tamaño de bloque N_b

La función inversa de DesplazarFila será, un desplazamiento de las filas de la matriz de estado el mismo número de posiciones que se indica en la tabla 3.6, pero en este caso a la derecha.

Función MezclarColumnas

Para esta función, cada columna del vector de estado se considera un polinomio cuyos coeficientes pertenecen al campo GF (2^8) — es decir, son bytes – y se multiplica módulo el polinomio ($x^4 + 1$) por el polinomio multiplicador siguiente:

$$c(x) = 03x^4 + 01x^2 + 01x + 02$$

donde 03 es el valor hexadecimal (un byte) que es un elemento de GF(2^8), en este caso 00000011, o sea, x + 1, y así sucesivamente.

La función inversa de MezclarColumnas se obtiene multiplicando ahora cada columna de la matriz de estado por el polinomio:

$$d(x) = 0Bx^4 + 0Dx^2 + 09x + 0E$$

3.5.2.2 Cálculo de las subclaves

Las diferentes subclaves K_i, necesarias para las rondas, se derivan de la clave principal K mediante el uso de dos funciones: una de expansión y otra de selección. Siendo n el número de rondas que se van a necesitar, la función de expansión permite obtener, a partir del valor de K, una secuencia de $4\cdot(n+1)\cdot N_b$ bytes. La selección posterior simplemente toma consecutivamente de la secuencia obtenida bloques del mismo tamaño que la matriz de estado, y los va asignando a cada subclave K_i.

Sea K (i) un vector de bytes de tamaño $4 \cdot N_k$, conteniendo la clave, y sea W(i) un vector de $N_b \cdot (n + 1)$ registros de 4 bytes, siendo n el número de rondas. La función de expansión tiene dos versiones, dependiendo del valor de N_k:

a) Si $N_k \leq 6$:

1. Para i desde 0 hasta $N_k - 1$ hacer
2. $W(i) \leftarrow (K(4 \cdot i), K(4 \cdot i + 1), K(4 \cdot i + 2), K(4 \cdot i + 3))$

3. Para i desde N_k hasta $N_b \cdot (n + 1)$ hacer
4. tmp ← W(i − 1)
5. Si *i* mód $N_k = 0$
6. tmp ← Sub(Rot(tmp)) $\oplus$ Rc(i/N_k)
7. W(i) ← W(i − N_k) $\oplus$ tmp

b) Si $N_k > 6$:

1. Para i desde 0 hasta $N_k - 1$ hacer
2. W(i) ← (K(4 · i), K(4 · i + 1), K(4 · i + 2), K(4 · i + 3))
3. Para i desde N_k hasta $N_b \cdot (n + 1)$ hacer
4. tmp ← W(i − 1)
5. Si *i* mód $N_k = 0$
6. tmp ← Sub(Rot(tmp)) $\oplus$ Rc(i/N_k)
7. Si *i* mód $N_k = 4$
8. tmp ← Sub(tmp)
9. W(i) ← W(i − N_k) $\oplus$ tmp

En los algoritmos anteriores, la función Sub devuelve el resultado de aplicar la caja-s de AES a cada uno de los bytes del registro de cuatro que se le pasa como parámetro. La función Rot desplaza a la izquierda una posición los bytes del registro, de tal forma que si se le pasa como parámetro el valor (a, b, c, d) este devuelve (b, c, d, a). Finalmente, Rc(j) es una constante definida de la siguiente forma:

- Rc(j) = (R(j), 0, 0, 0)
- Cada R (i) es el elemento de GF (2^8) correspondiente al valor $x^{(i-1),}$ módulo $x^8 + x^4 + x^3 + x + 1$.

3.5.2.3 Seguridad de AES

Según sus autores, es altamente improbable que existan claves débiles, o semidébiles, en AES, debido a la estructura de su diseño, que busca eliminar la simetría en las subclaves. También se ha comprobado que es resistente a criptoanálisis tanto lineal como diferencial. En efecto, el método mas eficiente conocido hasta la fecha para recuperar la clave a partir de un par: texto cifrado – texto claro, es la búsqueda exhaustiva, por lo que se puede considerar a este algoritmo como uno de los más seguros en la actualidad.

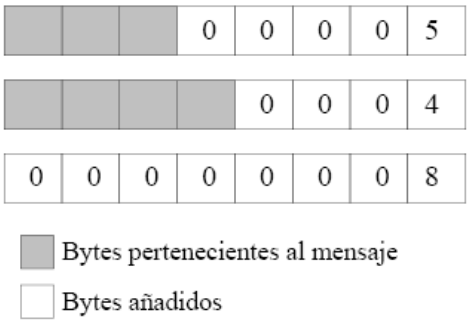


Figura 3-8 Relleno (padding) de los bytes del último bloque al emplear un algoritmo de cifrado por bloques

3.5.3 Otros algoritmos simétricos

En el proceso de búsqueda para encontrar un algoritmo más robusto y más seguro se han inventado un sin número de algoritmos de cifrado de bloques entre los que se pueden destacar BLOWFISH (creado por Schneier en 1994), Crab (ideado por Kaliski y Robshaw en 1994), FEAL (conceptualizado por Shimizu y Miyaguchi en 1988), KHAFRE (creado por Merkle en 1991), L0KI91 (ideado Brown y otros en 1991), NEWDES (conceptualizado por Scott en 1985), REDOCII (creado por Cusick y Wood en 1991), y SAFER K64 (diseñado por Massey en 1994). Para obtener más información de cada uno de estos algoritmos, se recomienda consultar la bibliografía [20]. Enseguida se describirán únicamente aquellos estrechamente relacionados con este trabajo de tesis.

3.5.3.1 Blowfish

Blowfish es un algoritmo de encriptado de bloque creado por Bruce Schneier [20]. Inicialmente pensado para construirse en "*hardware*", sus criterios de diseño se centraron en lo siguiente:

1. *Velocidad. Blowfish* encripta un byte en 26 ciclos de reloj, en un microprocesador de 32 bits.
2. *Memoria.* Sólo requiere 5 kB.
3. *Simplicidad. Blowfish* utiliza operaciones sencillas: suma, XOR y búsquedas en tablas de 32 operandos.
4. *Longitud de llave variable*. Por último, la longitud de la llave para *Blowfish* puede ser hasta de 448 bits.

Blowfish es un cifrador de bloque de 64 bits con una llave de longitud variable. El algoritmo consiste de dos partes: la expansión de la llave y el encriptado de datos. La expansión de clave convierte una clave de a lo más 448 bits en varios arreglos de claves totalizando 4168 bytes. La encriptación de datos consiste simplemente en una función iterada 16 veces. Cada iteración consiste de una permutación dependiente de la clave y una sustitución que depende tanto de la clave como de los datos.

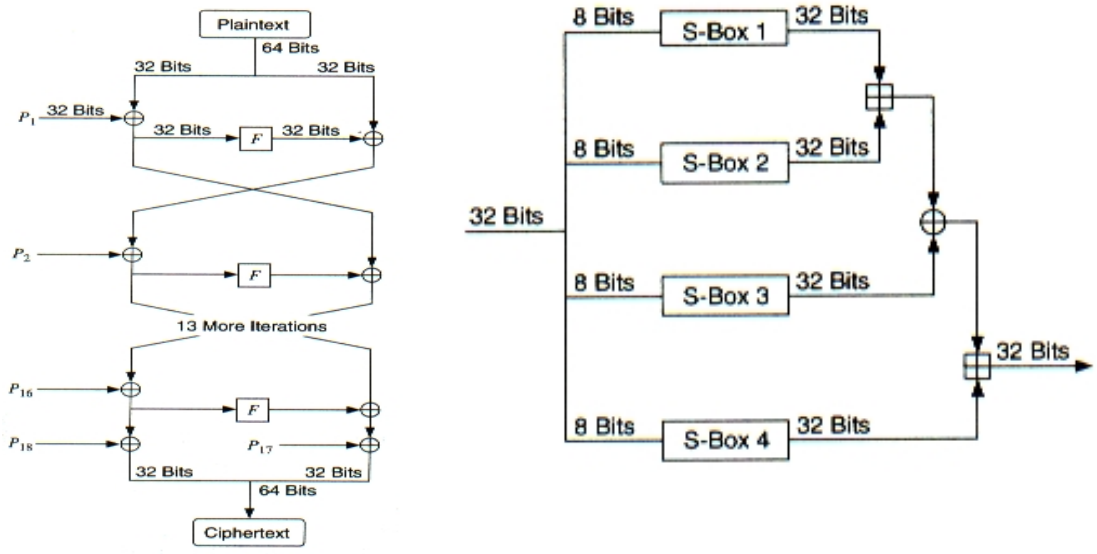


Figura 3-9. (a) Esquema general de Blowfish. (b) Detalle de una función F [20]

3.6 Uso de Funciones Resumen, o de Huella ("hash") para Criptografía

Una función resumen, o de "*hash*" *H* es una transformación que, tomando como entrada una cadena *x* de bits de longitud variable, produce como salida una cadena *h* de bits de longitud fija (*h = H(x)*). Para que una función de este tipo pueda usarse con propósitos criptográficos, se debe cumplir una serie de requisitos [38]:

- La entrada puede tener cualquier longitud. Deben proveerse mecanismos para evitar el desbordamiento ("*overflow*").
- La salida debe ser de longitud fija, independientemente de cual fuera la longitud de la entrada.
- Para cualquier entrada, su resumen (o valor de "*hash*") debe ser sencillo de calcular.
- La función resumen debe ser de un "único sentido", entendiendo por este concepto que, dado *f(x)*, debe ser computacionalmente difícil encontrar un valor *y* (tal vez el mismo *x*) tal que *f(y) = f(x)*.
- Es difícil encontrar dos entradas *x* e *y*, tales que *H(x) = H(y)* (colisiones).

Al resumen, o valor de "*hash*" de un mensaje *M* se le llama generalmente huella digital de *M*. Si la salida de la función tiene una longitud de *n* bits, entonces existen $k = 2^n$ salidas diferentes. Las funciones resumen son también extensivamente utilizadas como parte de los mecanismos que generan números aleatorios. Ejemplos de funciones resumen usadas en criptografía son los algoritmos denominados: MD2, MD4, MD5 o SHA-1.

3.6.1 Ataque de las funciones resumen

Como se desprende de las condiciones que debe respetar una función resumen, el ataque a dichas funciones puede verse desde dos puntos de vista.

Si, dado un mensaje *x* y su resumen *H (x)*, sólo mediante una búsqueda exhaustiva es posible hallar otro mensaje *y* tal que *H (x) = H (y)*, entonces, la función resumen *H* se denomina débilmente libre de colisiones. La búsqueda exhaustiva consiste en calcular el resumen de cada entrada posible, hasta obtener el valor *H(x)* conocido. Este ataque aparece en un escenario con dos actores, en el cual una tercera parte intenta engañar a una de las otras dos, calculando una entrada con un valor resumen igual al del mensaje cuyo resumen ha interceptado.

Una función resumen fuertemente libre de colisiones es aquella para la que no es posible hallar dos mensajes *x* e *y* tales que *H (x) = H (y)*. En este caso es una de las dos partes implicadas la que trata de engañar a la contraria, tratando de atacar la integridad de los mensajes cuyo resumen se ha calculado.

Generalmente, el ataque a funciones resumen aparece en el primero de los escenarios, en el que se utilizan huellas digitales para firmar un mensaje. Un típico ataque es el conocido como "ataque del cumpleaños", el cual se basa en una curiosa paradoja, que da nombre al ataque. Esta, establece que la probabilidad de que dos o más personas de un grupo de 23 compartan la misma fecha de cumpleaños es mayor de 1/2.

Matemáticamente, suponiendo una función, alimentada con una entrada aleatoria, que produce *k* salidas equiprobables. Entonces, probando repetidamente diferentes entradas, se espera obtener la misma salida después de probar $(2k)^{1/2}$ entradas y no, como cabria esperar $k^{1/2}$ [15].

Teniendo en cuenta que $k = 2^n$ (en donde *n* es la longitud en bits de la salida) se deben elegir valores de *n* lo suficientemente grandes como para impedir este cálculo inverso.

3.6.2 MD4 y MD5

MD4 y MD5 son funciones resumen usadas en criptografía. Su nombre proviene de "*Messages Digest*" (resumen de mensajes) y fueron diseñados por Ron Rivest. Se emplean fundamentalmente en la generación de huellas digitales de documentos, mensajes de correo electrónico y objetos similares. Los dos algoritmos generan huellas con una longitud de 128 bits [38].

MD4 fue introducida en 1990 con el objetivo fundamental de ser una función rápida. Sin embargo, ya en 1995 se demostró que era posible hallar colisiones para MD4 en menos de un minuto utilizando un simple PC [39]. Consecuentemente, MD4 ya no es considerada segura.

MD5 es una versión mejorada (aunque algo más lenta) de MD4, desarrollada por Ron Rivest en 1991. Su fortaleza es grande y, dado que la longitud de la salida es 128 bits, la probabilidad de obtener dos mensajes con el mismo resumen es de 2^{64}, en tanto que la dificultad de obtener un mensaje cuyo resumen sea igual a uno dado es de 2^{128} [39].

Aunque se ha avanzado en su estudio y se ha demostrado que es posible hallar colisiones para la función de compresión que utiliza el algoritmo, no se ha demostrado que puedan hallarse para el algoritmo entero. De momento es considerado seguro, aunque se recomienda que, se debe actualizar cualquier producto que lo utilice con otros algoritmos como SHA-1.

3.6.3 SHA y SHA-1

SHA ("*Secure Hash Algorithm*") fue desarrollado en 1993 por NIST ("*National Institute for Standards and Technology*") junto con NSA ("*National Security Agency*") en EE.UU. para su uso en la norma estadounidense de firma digital. En 1994, el propio NIST publicó una revisión de este último, conocida como SHA-1, la cual corrige un defecto no publicado de SHA.

SHA es muy similar en su modo de operación a MD5. Utiliza como entrada mensajes de hasta 2^{64} bits, procesa la información en bloques de 512 bits, para los que genera salidas de 160 bits, más largas que las producidas por cualquier otra función resumen utilizada anteriormente. Este algoritmo es ligeramente más lento que MD5, pero la mayor longitud del resumen del mensaje lo hace más seguro frente a la búsqueda de colisiones usando la fuerza bruta [38].

3.7 Criptografía de Clave Asimétrica o de Clave Pública

El principal problema que presenta el uso práctico de la criptografía de clave simétrica es la distribución de las claves. La criptografía de clave asimétrica o pública, sin embargo, usa claves diferentes para cifrar y descifrar un mensaje, aunque sus aplicaciones son diferentes. Lo único que se transmite de un usuario a otro es el mensaje cifrado. La clave para cifrar es pública, y para descifrar es privada. En las firmas digitales, estos roles están invertidos.

En 1976, Whitfield Diffie y Martín Hellman publican la idea de que cada usuario tenga dos claves: una pública (P_i, conocida por cualquiera) y otra privada (S_i, sólo conocida por su dueño).

Entre estas claves existe una relación particular que permite a una de ellas cifrar un mensaje mientras que la otra es empleada para descifrarlo. Las claves privadas deben ser conservadas por su propietario del modo más seguro posible.

Suponiendo un algoritmo de cifrado *E* y otro de descifrado *D*, aplicados a un mensaje *M.* Debe cumplirse que [16]:

$$D(E(M, P_i), S_i) = M$$

La seguridad de un sistema de este tipo depende de que las funciones de cifrado y descifrado, *E* y *D*, cumplan una serie de condiciones:

- Dados el mensaje *M* y la clave pública *P* que se vaya a utilizar, el mensaje cifrado, *C*, debe ser fácil de calcular.
- Dado *C*, el mensaje original, *M*, no debe ser obtenible de forma sencilla.
- Dados *C* y la clave privada, *S*, debe ser sencillo descifrar el mensaje original.
- Para que sea práctico el uso de criptografía de clave asimétrica, debe ser sencillo calcular parejas aleatorias de claves *P* y *S*.

La criptografía de clave asimétrica posee, sin embargo, dos inconvenientes:

1. El primero se refiere a la velocidad. Los sistemas basados en clave asimétrica son notablemente más lentos que sus contrapartes de clave simétrica (por lo general y como mínimo, dos órdenes de magnitud). Por tanto estos sistemas no suelen ser adecuados para el cifrado masivo de datos.
2. El segundo está relacionado con la validación de la clave. La discusión sobre la fortaleza de un algoritmo de clave asimétrica es irrelevante sin una discusión previa sobre el protocolo de validación de las claves.

3.7.1 Algoritmo Rivest-Shamir-Adleman (RSA)

El algoritmo RSA se basa en el hecho de que la factorización de números primos es un problema de resolución computacionalmente difícil. El algoritmo RSA está descrito en infinidad de libros y páginas *Web*; y su definición es la siguiente:

Primero es necesario calcular las claves:

a. Encontrar dos números primos grandes (de 100 cifras o más), ***p*** y ***q***.
b. Definir ***n*** (conocido como módulo) como: ***n = pq***
c. Definir ***z*** como: ***z = (p-1)(q-1)***
d. Encontrar un número primo aleatorio ***e*** menor que el módulo y tal que ***e*** y ***z*** sean primos entre sí.
e. Determinar un valor ***d*** tal que se cumpla que ***(ed - 1)*** es divisible entre ***z*** (***d*** existe y es único).

Considerando la clave pública como los valores *n* y *e*:
El cifrado del mensaje M se obtendrá según la siguiente operación: $C = M^e \pmod n$

Y con la clave privada *d*, el descifrado mediante la siguiente: $M = C^d \pmod n$

Por tanto, la clave pública estará constituida por el par ***(n,e)***, mientras que la clave privada es ***(d)***.

Suponiendo ***p=47*** y ***q=57***
Por tanto, ***n=pq=2773***

De estos datos, se calcula ***(p-1)(q-1)=2668***

Eligiendo ***e=17***, se calcula ***d*** utilizando algoritmos 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).

3.7.1.1 Funcionamiento

Como ya se ha señalado, el algoritmo RSA se apoya en el hecho de que factorizar números muy grandes es un problema computacional de difícil resolución. Si un intruso que quisiera romper el algoritmo fuese capaz de factorizar ***n*** (parte de la clave pública), entonces podría utilizar estos factores para deducir rápidamente ***d***. Por lo tanto, si fuese fácil factorizar números grandes, sería fácil romper RSA.

Para lograr la máxima seguridad, es necesario utilizar enteros de más de 100 dígitos de longitud, pues factorizar rápido números más pequeños es posible. Según Bruce Schneier [16], un número de 129 dígitos decimales está en el borde de las tecnologías y técnicas de factorización. Además, se debe asegurar que el producto ***(p-1) (q-1)*** no tiene factores primos pequeños.

Fruto de los requerimientos para hacer seguro RSA, surge su principal problema: su lentitud. En general, se elige como clave pública ***e***, 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 rápido que la generación de claves). El cifrado, que utiliza la clave pública, tiene una complejidad de $O(k^2)$, el descifrado $O(k^3)$ y la generación de claves $O(k^4)$, en donde ***k*** es la longitud en bits del módulo.

Una práctica habitual es elegir como clave pública un exponente pequeño, con el cual además la exponenciación binaria sea rápida, típicamente 1001 ó 10001, variando el módulo.

Clave pública	***n*** : producto de dos números primos, ***p*** y ***q*** (que deben permanecer secretos).
	e : relativamente primo a ***(p-1)(q-1)***.
Clave privada	$d : e^{-1} \bmod ((p-1)(q-1))$
Cifrado	$c = m^e \bmod n$
Descifrado	$m = c^d \bmod n$

3.7.2 Algoritmo ElGamal

El algoritmo ElGamal (también conocido como algoritmo Diffie-Hellman, variante ElGamal) se basa en la dificultad de calcular algoritmos discretos en un campo finito.

Para generar un par de claves, se elige un número primo, *p*, y dos números aleatorios, ***g*** y ***x***, de modo que sean más pequeños que ***p***. Calculando:

$$y = g^x \bmod p$$

La clave pública es ***y***, ***g*** y ***p*** (grupos de usuarios pueden compartir ***g*** y ***p***, diferenciándose sólo en ***y***). La clave privada es ***x***. Este algoritmo se puede utilizar tanto para cifrado, como para firmas digitales. Dado que el mismo no fue utilizado directamente en esta tesis, dejamos al lector consultar [15] para mayor detalle.

3.8 Autentificación y firma digital

La criptografía de clave pública puede ser utilizada para identificar sin ambigüedades al remitente de un mensaje. Esto es posible teniendo en cuenta que, si el remitente encripta con su clave privada el mensaje que envía, éste solamente puede ser desencriptado en el destino utilizando la clave pública del remitente y por lo tanto se puede verificar la firma.

La probabilidad de que dos personas diferentes tengan la misma combinación clave pública / clave privada es insignificante.

La desventaja de la utilización de cualquier algoritmo de clave pública es su lentitud, por lo que resulta poco práctico el cifrado asimétrico del mensaje entero. Frecuentemente se utilizan más bien para cifrar las claves de los algoritmos simétricos y poder ser así enviadas al receptor de manera segura.

3.8.1 Códigos de integridad

Como se acaba de señalar, resulta poco práctica la aplicación de técnicas asimétricas a un mensaje entero. En tal caso, se utilizan funciones resumen que derivan una huella digital (en ingles, MAC, "*Messages Authentication Code*") a partir de un cierto volumen de datos. [38]

Esto es debido a que las funciones resumen poseen dos propiedades que las hacen ideales para este trabajo. La primera es que su resultado es relativamente corto (típicamente una huella tiene entre 128 y 160 bits). Segundo y más importante, aunque sea teóricamente posible encontrar dos mensajes con idéntica huella, la probabilidad de que esto ocurra es ínfima. Si se manipulan los datos, la huella cambia.

3.8.2 Firmas digitales

La firma manuscrita como medio para acreditar la identidad del firmante de un documento ha sido, y sigue siendo, ampliamente usada por la sociedad desde hace siglos. Su equivalente en la actual sociedad de la información es lo que se conoce como firma digital.

Además de la capacidad de autentificar al signatario de un mensaje, la firma digital posee otra cualidad interesante, que consiste en mantener la integridad del mensaje firmado. Dado un mensaje, basta calcular su huella digital y cifrarla con la clave privada del remitente para obtener simultáneamente la seguridad de que el contenido no se manipula (integridad), y de que el firmante es quien dice ser (autenticación). El protocolo que se utiliza es el siguiente:

1. Se aplica una función resumen ***f*** al mensaje ***M***. El resultado ***f(M)*** es la huella digital del mensaje.
2. A continuación el resultado del paso anterior, se encripta con la clave privada.
3. Posteriormente es enviado tanto el mensaje como la firma digital (la huella digital cifrada con su clave privada) al receptor.
4. El receptor, que conoce la función resumen utilizada y la clave pública del emisor, realiza dos operaciones: aplica la función resumen al mensaje y descifra la firma digital.
5. Si ambos resultados coinciden, el receptor tiene la certeza de dos hechos: que el mensaje no ha sido modificado en su tránsito por la red; y que el mensaje ha sido emitido, efectivamente, por quien dijo ser el emisor.

El problema ahora es de otro tipo. ¿Cómo asegurar que el par clave pública / clave privada que se asocio al emisor es realmente suya y no de un intruso? Es el problema de validación de la clave.

3.9 Certificados Digitales

Los certificados son documentos digitales que atestiguan que una clave pública corresponde a un individuo, o entidad determinados. De este modo se evita 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. Adicionalmente, puede contener la fecha de expedición del certificado, la de expiración de la clave, el nombre del notario electrónico que emitió el certificado y un número de serie. De todo ello calcula la huella digital con la función de "*hash*" adecuada y la cifra con su clave privada.

Los certificados pueden adoptar múltiples formas. El formato más difundido está definido por la norma del ITU-T X.509 (versión 3), la cual forma parte del servicio de directorio diseñado por ISO para el modelo OSI. En el certificado se incluyen [38]:

- La versión de la norma X.509 usada.
- El número de serie del certificado.
- El algoritmo utilizado por la autoridad de certificación (algoritmo de clave asimétrica y función de resumen usada).
- Los nombres que identifican unívocamente al dueño del certificado y a la autoridad de certificación.
- La clave pública del dueño del certificado, junto con la información de los algoritmos utilizados.
- La firma digital de la autoridad de certificación.

Las autoridades de certificación deben ser entes fiables y ampliamente reconocidos que firman (con conocimiento de causa y asunción de responsabilidades legales) las claves públicas de las personas, rubricando con su propia firma la identidad del usuario. El destinatario de un mensaje no recibe la clave pública del remitente sino su certificado.

Debido a la posición comprometida que ocupan las autoridades de certificación, éstas deben tomar extremadas precauciones para evitar que sus claves caigan en manos de intrusos, lo cual comprometería todo el sistema. Para ello tendrá que utilizar claves largas y dispositivos especiales para su almacenamiento.

Además, cuando emiten un certificado, deben estar seguros de que lo hacen a la persona adecuada. No se puede 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.

3.9.1 Ciclo de vida de una clave

Las claves deben tener una fecha de expiración. De esta forma, es más difícil que los algoritmos que las utilizan sufran algún ataque. Cuando una clave ha sido averiguada por intrusos, se dice que ha sido comprometida.

El ciclo de vida de una clave incluye los siguientes periodos [38]:

a) *Generación* y, quizá, *registro* de la clave, o par de claves. La clave, o par de claves deben ser generadas por su propietario, o por la entidad que vaya a utilizarlas. Un problema frecuente radica en que los algoritmos generadores de claves (aleatorias) no son suficientemente "buenos". Cuando la clave es utilizada con algoritmos 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.

3.9.2 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 permitan acceder al sistema mediante el "*hardware*" adecuado.

3.9.3 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 su seguridad. Sin embargo, la presión de la industria informática de los EE.UU. es fuerte y se vislumbra una relajación de las restricciones. La contrapartida sería 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) [38]:

1. La *custodia de claves ("key escrow")*: es el usuario u organización quien genera su clave, o claves, y las entrega a otra parte que las 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 varias manos.
2. La *tercera parte de confianza ("trusted third-party")*: 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.

3.10 Intercambio de claves simétricas (sistemas híbridos)

Dado que los algoritmos criptográficos de clave pública (como el ya citado RSA) son lentos, se han desarrollado sistemas híbridos que utilizan criptografía de clave simétrica y de clave pública. El mecanismo, a grandes rasgos, es el siguiente:

1. Se genera una clave aleatoria que servirá de clave a un algoritmo simétrico (por ejemplo, AES).
2. Esta clave es cifrada con la clave pública del receptor y es enviada (criptografía de clave asimétrica).
3. El receptor toma el mensaje y, con su clave privada, procede a descifrarlo, obteniendo así la clave para el algoritmo simétrico.
4. El resto de comunicaciones entre el emisor y receptor se lleva a cabo usando algoritmos simétricos con la clave transmitida.

Con estos métodos híbridos se eliminan los problemas que origina la distribución de claves. Ahora bien, aparecen problemas nuevos, debidos fundamentalmente a la posibilidad de suplantación de alguno de los interlocutores. Esto solamente puede evitarse intercambiando certificados en lugar únicamente de claves cifradas.

3.10.1 Algoritmo de intercambio de claves Diffie-Hellman

Este algoritmo permite que dos usuarios intercambien una clave a través de un medio inseguro. Se utilizan dos parámetros públicos, p y g. El primero de ellos, p, es primo. El segundo, conocido como generador, se define como:

$$\forall\, n \in [1..(n-1)]\ \exists\, x \,/\, n = x^g \pmod{p}$$

El funcionamiento del protocolo se describe a continuación. Cuando dos entes, llamémosles “Pepe” y “Manolo” desean intercambiar una clave llevan a cabo el siguiente proceso:

I. Pepe genera aleatoriamente un valor privado a.
II. Manolo hace lo propio, generando b.
III. Pepe genera un valor público $g^a \bmod p$.
IV. Manolo hace lo mismo generando $g^b \bmod p$.
V. Manolo y Pepe intercambian estos valores públicos.
VI. A continuación, Pepe calcula $K_a = (g^b \bmod p)^a = g^{ab} \bmod p$.
VII. Manolo hace un cálculo similar: $K_b = (g^a \bmod p)^b = g^{ab} \bmod p$.
VIII. En este momento, ambos poseen una clave común K:

$$K = K_a = K_b = g^{ab} \bmod p$$

El fundamento de este algoritmo consiste en que es difícil calcular K, aún conocidos los valores públicos p, g, $g^b \pmod p$ y $g^a \pmod p$.

Como ya se ha comentado para el caso general, el inconveniente principal de que adolece este algoritmo es la posibilidad de que un intruso intercepte las comunicaciones entre los dos interlocutores. Aunque aún con los valores interceptados, el intruso

necesitaría calcular un logaritmo numérico (el cual es difícil de calcular), para tener mayor protección será necesario, simplemente, utilizar certificados.

3.11 Fortaleza de un algoritmo criptográfico

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 poco tiempo (aunque teóricamente pueda serlo). Sin embargo, esto no es, a menudo, fácil de demostrar. Un algoritmo criptográfico es considerado seguro si [38]:

- 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 posible es lo suficientemente grande como para que la búsqueda exhaustiva no sea práctica.

Teóricamente, cualquier algoritmo 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 algoritmos de clave simétrica, por ejemplo, una clave de 32 bits de longitud requiere 2^{32} operaciones (aproximadamente 10^9).

Existe una regla empírica, conocida como la Ley de Moore, que establece que la potencia de computación disponible para una inversión monetaria fija se duplica, 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. 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 costo derivado de usar claves seguras (de 128 o más bits de longitud) no es significativamente mucho mayor que en el que podrían incurrir cifrados con claves "débiles".

En el caso de los algoritmos 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 pública. En el caso del algoritmo RSA, esto es equivalente a factorizar un entero muy grande con dos factores primos también grandes. Lo cual computacionalmente es difícil en la actualidad.

CAPÍTULO 4

IMPLEMENTACIÓN DE LA SEGURIDAD DE INFORMACIÓN PARA LA CREACIÓN DE UN SITIO DE COMERCIO ELECTRÓNICO

Implementación de la Seguridad de Información para la Creación de un Sitio de Comercio Electrónico

En este capítulo se explicará la implementación de un Sitio *Web* por medio del cual se presente explícitamente un ambiente de comercio electrónico con todas las características y circunstancias por las cuales requiere la protección de su información; así como los análisis y consideraciones necesarios para determinar y diseñar las soluciones correctas.

Para la realización del portal se utilizó el Modelo General del cual se habló en el capitulo 2 a fin de manejar tanto el modo B2B como el B2C. Este escenario puede superar a un sitio de venta propiedad de la Casa de *Software* al manejar los productos de diferentes Casas de *Software* y tener la capacidad de crear CD-ROMs personalizados de diferentes autores.

4.1 El Negocio a Desarrollar

Se decidió implementar un sitio de comercio electrónico enfocado a la venta de música digitalizada, el cual vende sus canciones (mercancía) totalmente en formato bit ya que ésta puede ser transmitida al comprador por una red de datos. Es decir, que el usuario final selecciona, compra y descarga las canciones, todo por medio de Internet, y cuyo mercado es toda persona con una o más computadoras con acceso a Internet, y que posea una cuenta bancaria. Como ejemplificación de dicho mercado, se tiene la ciber-comunidad de usuarios de las redes P2P (peer-to-peer) de igual a igual que buena parte de los archivos compartidos en estas redes son de música y vídeo; y esto ha llevado a muchos observadores, entre ellos la mayor parte de las empresas discográficas y distribuidoras, a concluir que estas redes suponen una gran amenaza a los modelos empresariales tradicionales ya establecidos.

4.1.1 Motivación del negocio

La razón de enfocar la creación de un sitio de comercio a la venta de música se justifica por las siguientes razones:

- El actual desarrollo y despliegue de tecnologías.
- La creciente demanda de las empresas discográficas por regularizar el intercambio de canciones de los modos P2P.

Las actuales velocidades de transferencia de las conexiones a Internet provistas para domicilios nacionales.

La variable “velocidad de transferencia” es de gran importancia al evaluar cualquier negocio electrónico desde la perspectiva del cliente. Ya que en la actualidad la mayoría de los posibles consumidores tienen conexiones relativamente lentas, lo cual dificulta la transferencia de archivos grandes como son: videos, películas, videojuegos etc. (Ver tabla 4.1)

Tipo de Información	*Tamaño promedio (1 Byte = 8 bits)*	*Tiempo de transferencia máximo a tráfico máximo (1 kbps)*	*Tiempo de transferencia mínimo a tráfico máximo (4 kbps)*	*Tiempo de transferencia mínimo a tráfico mínimo (42 kbps)*
Página *Web* (con algunas imágenes pequeñas y un sonido de fondo)	60 kBytes	480 segundos = 8 minutos	120 segundos = 2 minutos	11.4 segundos
Canción digitalizada	4 min = de 40 kBytes a 4 Mbytes según el tipo de formato utilizado	320-32,768 segundos = 5.33-546.13 minutos	80-8192 segundos = 1.33-136.53 minutos	7.6-780.2 segundos
Video musical digitalizado	4 min = 40 Mbytes, aunque también varía según el tipo de formato utilizado	327,680 segundos = 5461 minutos = 91 horas	81,920 segundos = 1365 minutos = 22.75 horas	7801 segundos = 130 minutos =2 horas y 10 min

Tabla 4-1.Comparación de tiempos de transferencia

Sin embargo si bien este sitio no esta enfocado a la venta de archivos demasiado grandes esto es debido principalmente a la velocidad de transferencia ya que al momento de implementar la seguridad en el portal electrónico y a la hora de realizar las aplicaciones criptográficas en que se basa este modelo no hay ninguna variación en cuanto al tamaño del archivo.

Para la criptología, esto constituye la puerta a un mercado mucho más extendido al ocultamiento de los secretos gubernamentales ultra-confidenciales; se convierte en una herramienta más popular al ciudadano, al permitir:

1. La protección de bienes de uso y/o consumo común (dinero, música, videos, documentos, informes, planos); y
2. La protección de las relaciones personales y/o empresariales factibles y/o motivadas por la existencia del ciberespacio.

Y en este caso particular, se crea un negocio con un producto menos propicio a la "piratería".

4.1.2 Principales entes a proteger

Con este panorama general y lo expuesto en el capítulo anterior cabe recordar la pregunta: "¿qué se debe proteger en el comercio electrónico?"; se puede establecer rápidamente que lo más importante a proteger en este modelo es: el dinero y la música digitalizada.

4.1.2.1 Dinero

Necesidades de protección:

1. Se le debe proteger porque es un producto intermedio al permitir la compra de otros bienes y servicios. Su protección es indispensable por sus enormes

capacidades de ser usado varias veces y de satisfacer deseos humanos. Y es el único bien que se intercambia entre todas las personas físicas y morales de este modelo: Tienda Virtual, Casa Musical, Banco y Usuario Final.

2. Aunque en esta tesis no se pretende manejar efectivo digital (que constituye un posible substituto al actual efectivo en papel moneda); sí se manejan los bienes informáticos correspondientes a los actuales sistemas de venta electrónica, como son: los números de tarjetas bancarias (de crédito o débito). Por lo cual se deben proteger dichos bienes intermedios (e implícitamente al dinero) en formato bit (información capaz de ser transmitida en forma electrónica) ya que esto nos permitirá manejar el dinero en *Internet*. En consecuencia, los pricipales datos a proteger son:

    a) El número de cuenta bancaria;
    b) El número de la tarjeta asociada a dicha cuenta (tipo débito, crédito, internacional, etc); y
    c) La autorización electrónica relacionada a una transacción (actualmente su uso es poco frecuente en el modo B2C, aunque lo es más en el B2B).

4.1.2.2 Música

Necesidades de protección:

1. Al igual que el dinero, la música digitalizada puede ser utilizada varias veces y constituye un satisfactor humano. Por ende este bien se debe proteger por las siguientes dos características:

    a) Es un bien intermedio, ya que al vender la música digitalizada se puede obtener dinero. Y si se revende ilegalmente (piratería), el beneficio es distinto para los creadores y/o inversionistas del bien original.

    b) Es un bien final, para su consumo público. Lo cual significa que puede ser utilizada por cualquier persona con una computadora multimedia o reproductor musical adecuado; es decir, existe un mercado potencial de millones de consumidores distribuidos por todo el orbe.

2. Los hechos anteriores obligan a que la información (música) deba ser protegida: en primer plano ella misma; y en segundo plano, se deberá proteger también aquellos entes con quienes tiene alguna relación durante su ciclo de vida (a grandes rasgos: creación, almacenamiento, transporte y reproducción).

Por sí misma debe tener una protección en su estructura, lo cual impida ser "pirateada" por cualquier persona, y para ello se requiere de herramientas criptográficas, con el objetivo adicional de particularizar cada canción a las características de cada comprador.

4.2 Estructura y Modo de Operación del Portal

4.2.1 Estructura del Portal

En la actualidad, la mayoría de los sitios *Web* (comerciales o no), se diseñan, mantienen y actualizan con una gran variedad de herramientas de "*software*" y lenguajes de programación, cada uno con una serie de funcionalidades, fortalezas y debilidades que, aprovechadas en conjunto, facilitan el trabajo del diseñador.

Para ilustrar mejor donde se usa cada herramienta de seguridad, primero se necesita analizar la estructura de las páginas que conforma el sitio *Web*, para después entender los objetivos que debe cumplir el programa.

Para comenzar, se definirá la estructura del portal electrónico:

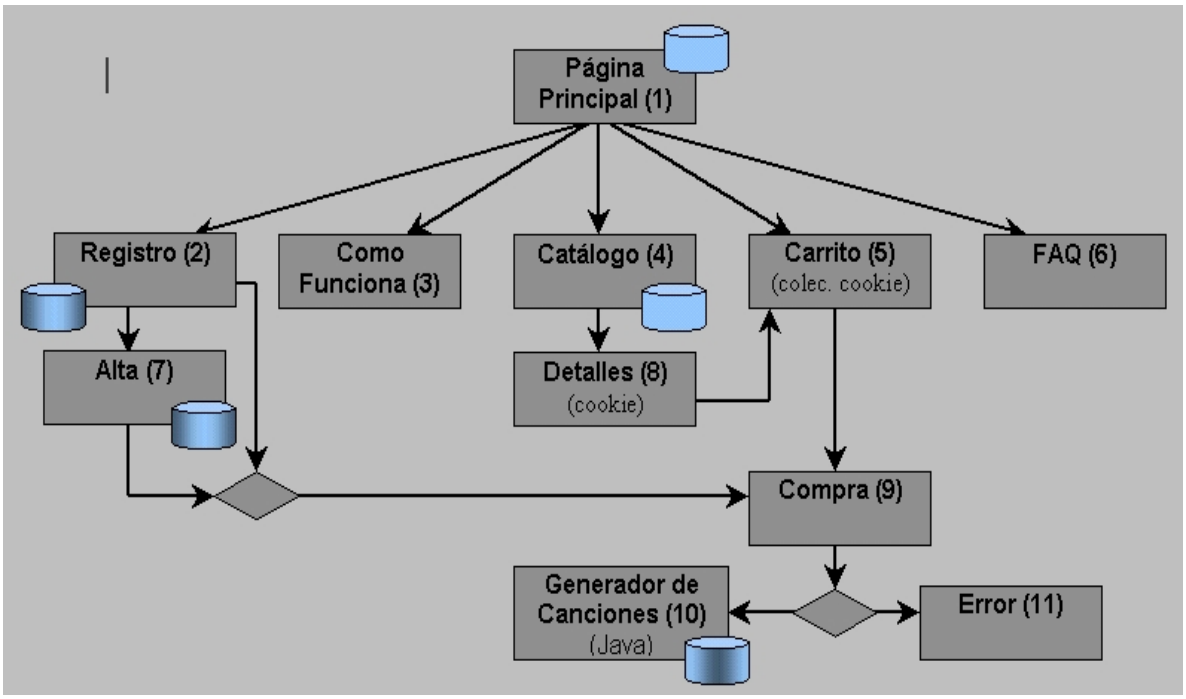


Figura 4-1. Jerarquía de Páginas Web


Base de Datos de Productos


Base de Datos de Usuarios


Base de Datos de Transacciones

No.	Descripción
1	En esta página se muestra en un menú el resto de las secciones del portal, un recuadro para que los usuarios registrados se den de alta y las "Ofertas de la Semana".
2	Página en la que el usuario puede darse de alta, solicitar su registro al sistema

	o pedir su "*password*" en caso de que lo haya olvidado. Además, en esta página se indica el correo electrónico de atención a clientes.
3	Se da una explicación detallada de la forma en la que se deberán realizar las compras, así como el mecanismo mediante el cual se reproducen las canciones en la computadora del usuario.
4	En esta página se despliegan las canciones (productos o mercancía a la venta) almacenadas en nuestra base de datos. En esta página se dan los detalles elementales de la canción.
5	Esta página despliega aquellos productos seleccionados por el usuario. Al inicio se encuentra vacío. El proceso de "llenado" se realiza a través de una "*cookie*".
6	Preguntas más frecuentes. En esta sección se dan *tips* y consejos para el uso de tarjetas de crédito en Internet, y se responden algunos cuestionamientos relativos al "*software*".
7	En esta página los usuarios que no están registrados en la base de datos de usuarios pueden crearse una cuenta nueva a través de la cual realizarán sus compras.
8	Cuando un usuario selecciona una canción, esta se almacena temporalmente en una "*cookie*", misma que sirve para desplegar el contenido completo del registro almacenado en la base de datos. Si el usuario decide comprar esta canción, primero tendrá que agregar los datos de la "*cookie*" temporal a la "*cookie*" que guarda los registros del carrito.
9	La página de compra pedirá el número de tarjeta de crédito. A continuación hará la verificación de que se trata de un número válido y posteriormente captura el número de folio de la transferencia.
10	Por último, nuestro portal tendrá una página en la que, primero, se desencriptarán los archivos "mkr" (provenientes de la Casa Musical) y se encriptarán con la llave propia del usuario. Después se generará la factura y por último, se enviará todo este paquete al usuario.
11	Página de error, en caso de que ocurra algún incidente durante la transferencia.

Tabla 4-2.Descripción de la pagina Web

4.2.2 Modo de Operación

El funcionamiento del sistema de seguridad, al que se llamara "*CryptoPlayer*", se basa en la existencia de dos componentes de "*software*": uno del lado del servidor ("*CryptoMusicMaker*") y otro más del lado del cliente ("*CryptoPlayer*").

"*CryptoMusicMaker*" es un sistema de almacenamiento, distribución y venta de música digitalizada, cuyo objetivo es poner a disposición del público una gran cantidad de títulos y temas musicales, permitiendo que sean los usuarios quienes personalicen su propia audioteca y únicamente compren aquellas melodías que sean de su interés personal.

La razón por la que se escogió el mercado de audio digital fue básicamente:

- Porque el audio digital es un bien informático, lo cual implica que para su venta y distribución no es necesario un soporte físico; por lo tanto, para mantener su valor, se requiere de un mecanismo que lo defienda contra intrusos y no debe permitirse su copia ilegal (con ayuda del cifrado).

El procedimiento mediante el cual un cliente cualquiera puede hacer uso del sistema es el siguiente:

a. Darse de alta en el sistema.
b. Se le proporcionará un "*password*".
c. A continuación deberá descargar "*CryptoPlayer*".
d. Durante la instalación del reproductor, éste solicitará el "*password*" que le fue proporcionado en el punto dos.
e. Ir al Catálogo
f. Agregarlo productos al "Carrito de Compras"
g. Comprar melodías
h. Proporcionar número de tarjeta de crédito
i. Si la operación se lleva a cabo con éxito, deberá descargar en su computadora el(los) tema(s) musical(es) de su "carrito" junto con su factura digital.

4.3 Resumen de Aplicaciones Empleadas

En esta sección se describirán las principales aplicaciones empleadas para el desarrollo del sitio de comercio electrónico realizado en esta tesis, así como su implementación.

4.3.1 PWS ("Personal Web Server")

Personal Web Server es un producto de *Microsoft*. Se trata de un servidor *Web* de escritorio que permite publicar páginas HTML y compartir documentos en una red corporativa desde equipos de escritorio. Además, PWS puede ser usado como plataforma de desarrollo para crear y probar sitios *Web* antes de ser cargados en el servidor de un proveedor de servicios integrados.

PWS se utilizo para probar el sitio antes de alojarlo en el servidor definitivo; dado que permite comprobar los vínculos, formularios, secuencias de comandos y aplicaciones, para asegurarse de que su apariencia y funcionamiento sean los correctos [32], y así posteriormente ser sustituido por el MIIS.

Figura 4-2. Página principal del PWS

4.3.2 IIS ("Internet Information Server")

Es un servidor *Web* de *Microsoft* originalmente diseñado para correr en la plataforma de Windows NT: Windows 2000 Profesional o Windows 2000 y 2003 Server, así como Windows XP, también en sus versiones *Profesional* y *Server*.

El Servicio de *Internet Information Server* (IIS) 5.1 ofrece la eficacia de las páginas *Web* a *Windows*. Con ayuda de IIS, se podrán compartir fácilmente archivos e impresoras, o bien se podrán crear aplicaciones para publicar de forma segura información en el *Web* a fin de mejorar la forma en que se comparte información. IIS es una plataforma segura para crear y distribuir soluciones de comercio electrónico y aplicaciones críticas en el *Web* [32].

IIS integra estándares de *Internet* consolidados con *Windows*, de tal forma que utilizar el *Web* no signifique tener que empezar desde el principio y aprender nuevas formas de publicar, administrar, o desarrollar contenido.

4.3.2.1 Instalación de IIS en *Windows*

Estas normas de instalación son aplicables, a nivel general, a las que se pueden encontrar en las distintas versiones de los sistemas operativos comentados antes, aunque en este documento se ha tomado *Windows* XP profesional para relatar los pasos y tomar las imágenes de las pantallas, esto fue debido a que se utilizo este sistema operativo como servidor para publicar la pagina *Web*.

MIIS se puede encontrar en el propio CD de instalación de *Windows*. Hay que acceder a la opción de "Instalar componentes opcionales de *Windows*" para poder cargarlo en el sistema. Para ello se tienen dos opciones:

1. Insertar el CD de instalación de *Windows* y en la ventana de autoarranque que se muestra, seleccionar la opción que permite "Instalar componentes opcionales de *Windows*"

2. En el Panel de control, se selecciona la opción de "Agregar o quitar programas" y en la ventana que aparece, se pulsa sobre el *icono* de la izquierda marcado como "Seleccionar o quitar componentes de *Windows*".

Una vez que se selecciono cualquiera de las dos opciones, se muestra la ventana para seleccionar los componentes adicionales de *Windows* que hay disponibles. En la lista, se marca la opción "Servicios de *Internet* Information Server (IIS)". Por defecto se seleccionan unos cuantos componentes, dentro de los que ofrece la instalación de IIS.

Al seleccionar la casilla de *Internet Information Server*, y apretar el botón de “Detalles”, se puede acceder a los componentes específicos que se desean instalar con el IIS, a continuación se muestran con un poco de detalle cuales son todas estas opciones:

- **Archivos comunes:** archivos necesarios para los componentes de *Internet Information Server*.
- **Complemento de servicios de *Internet Information Server*:** sirve para administrar el IIS.
- **Documentación:** documentación necesaria para profundizar en el funcionamiento del IIS.
- **Extensiones de servidor de *FrontPage2000*:** estas extensiones permiten que nuestro servidor pueda incluir formularios, contadores, etc.
- **Servicio de protocolo de transferencia de archivos (FTP):** solo necesario si queremos un servidor FTP.
- **Servicio SMTP***: Simple Mail Transfer Protocol* (SMTP), nos permite montar un servicio de mail dentro de nuestra intranet.
- **Servicio *World Wide Web*:** necesario para poder montar nuestro servidor de páginas Web.

Una vez instalados los componentes deseados, se aprieta el botón de "Siguiente" para comenzar la instalación, que se alargará unos minutos.

Acceso al servidor Web

Para poder acceder al servidor Web y así comprobar si se ha instalado correctamente IIS, simplemente se debe escribir http://localhost en Internet Explorer y debería aparecer una página *Web* informando que IIS está correctamente instalado. Además, aparecerá la documentación de IIS en una ventana emergente, si es que fue instalada.

4.3.2.2 Administración de IIS

Para administrar el servidor *Internet Information Server* en *Windows*, se dispone de un panel de control llamado "Servicios de Internet *Information Server*" al que se puede acceder de varias maneras.

1. Pulsando con el botón derecho en “MI PC” y seleccionando la opción que pone "Administrar". Esto abre "Administración de equipos". En la lista de la izquierda, en la parte de abajo aparece "Servicios y aplicaciones", entre los que se encuentra la opción: "Servicios de Internet *Information Server*"
2. Se puede acceder desde el panel de control. Si se tiene configurada la vista clásica se encontrara un icono que pone "Herramientas administrativas" y haciendo doble clic, se encontrará el icono para administrar IIS. Si se tiene

configurada la vista por categorías del panel de control (la que aparece por defecto en *Windows* XP) la búsqueda de la opción es un poco más compleja: Se selecciona "Rendimiento y mantenimiento" y dentro ya se encuentra el icono de "Herramientas administrativas", al que se accede al hacer doble clic para encontrar, entre otros, el icono para acceder a "Servicios de Internet *Information Server*".

3. Otra manera de acceder aparece en la ayuda de Internet *Information Server*. Se trata de hacer una búsqueda del archivo llamado "inetmgr.exe". Una vez localizado se puede ejecutar y aparece la consola de administración de IIS. Si se desea, se puede hacer un acceso directo a dicho archivo para no tener que buscarlo cada vez que se desee ejecutar.

Una vez que se ha accedido al panel "Servicios de Internet *Information Server*" se tiene la posibilidad de configurar al servidor *Web* en muchos aspectos, por ejemplo se puede, definir el documento “por defecto”, crear directorios virtuales, modificar las opciones de seguridad, etc.

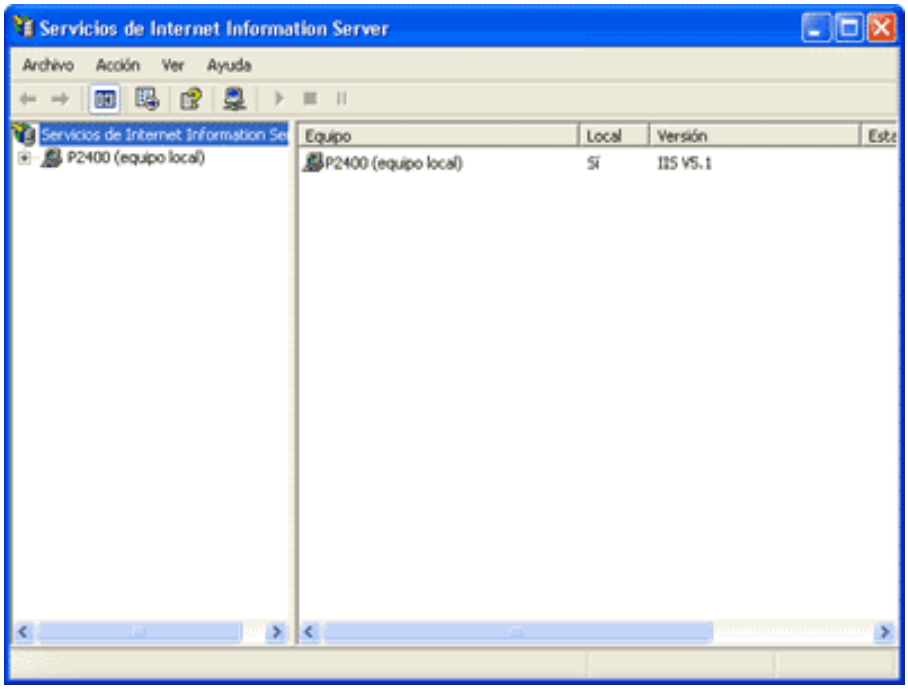


Figura 4-3. Panel de administración de IIS

“Por defecto” el nombre del sitio *WEB* es "Sitio *Web* Predeterminado", en el cual se puede cambiar el nombre en cualquier momento, simplemente pulsando dos veces en "Sitio *Web* predeterminado" y se puede modificar.

En la figura 4-3 se ilustran algunas de las opciones más generales para poder montar un servidor de páginas *WEB*. Haciendo clic con el botón derecho sobre "Sitio Web Predeterminado" y seleccionando "Propiedades", se muestran las opciones avanzadas de administración.

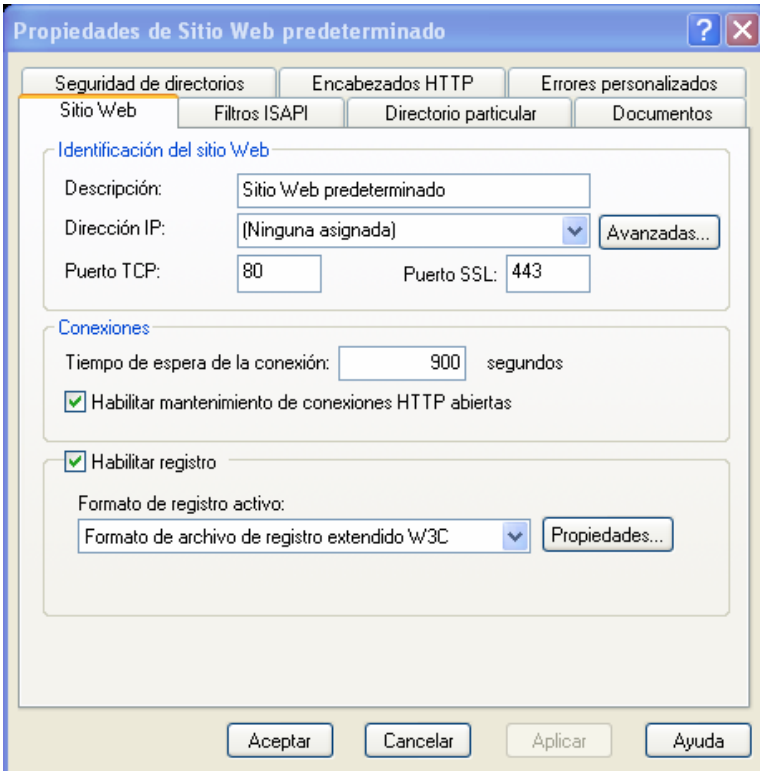


Figura 4-4. Panel de administración de IIS

Ahí se pueden cambiar diversas configuraciones, como por ejemplo, la dirección IP, la instalación del certificado del servidor (ver la siguiente sección), el directorio en donde se encuentra nuestra información de la pagina *Web*, etc.

4.3.2.3 Certificado de Servidor con IIS

La instalación de un certificado de servidor permitirá establecer conexiones seguras a través de *Internet*, de forma sencilla y transparente, tal y como se menciono en el capítulo anterior. Aunque para el presente trabajo se realizo un cerificado de servidor, este certificado no valdría para uso comercial, debido a que el propósito de un certificado de servidor es que el usuario pueda comprobar la identidad del servidor. No sirve de nada que el servidor pueda hacerlo si el usuario no puede. Al entrar un usuario al sitio *Web*, el servidor les dará la cadena de certificación y el usuario comprobara:

- Que la maquina tiene la clave privada del certificado de servidor.
- Que el nombre del certificado coincide con el nombre de tu dominio.
- Que el certificado de servidor no ha caducado.
- Que el certificado de servidor no esta revocado (opcional).
- Que el certificado de la CA que lo firmó no ha caducado ni esta revocado.
- Que dicho certificado raíz está en "SU" máquina, si no está, le saltará una pantalla en la que se advertirá si se desea confiar o no en este certificado.

Por esta razón el certificado de servidor tiene que estar firmado por uno de los certificados raíces que vienen por defecto instalados en la mayoría de los navegadores, es decir un certificado comercial, (de pago). Si todos los usuarios que van a entrar a la pagina *Web*

pertenecen a la misma LAN o a una empresa, se podrá distribuir el certificado raíz de tu CA para evitar esta pantalla.

Instalación del certificado en IIS

Para la instalación del certificado se tiene que generar una pareja de claves y un identificador. Después se prepara la petición, que solo incluirá el identificador y la clave pública -¡Nunca la privada!-.

Ahora se abre la consola del IIS como se menciono anteriormente, una vez abierta se abre la ventana de propiedades del Sitio *Web* Predeterminado en su menú contextual; se abre el "Asistente de certificado de Servidor *Web*" en la pestaña "Seguridad de Directorios", se pulsa en "Certificado de Servidor" y se pulsa “siguiente”.

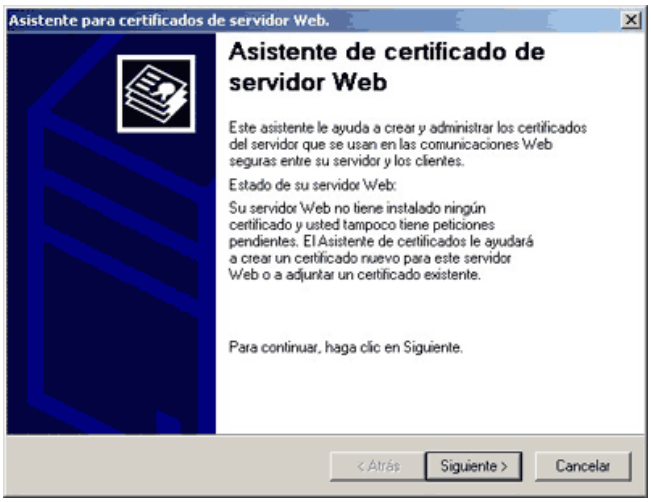


Figura 4-5. Asistente de certificado

Una vez terminado de configurar las características del certificado de servidor, se crea un archivo de nombre certreq.txt; el cual una vez que se tiene, junto a nuestra documentación, se procede a enviarlo a una autoridad certificadora y esta devolverá un certificado.

Figura 4-6. Ejemplo de certificado de servidor

Para utilizar el certificado de servidor, primero se tendrán que instalar los certificados raíces de las autoridades certificadoras raíz y secundarias. En el caso de este certificado

de prueba, solo el de la raíz, ya que no hay secundarias. Este certificado raíz será enviado junto con el certificado del servidor.

Una vez con el certificado se procede a "Instalar Certificado" y se sigue las instrucciones en pantalla aceptando todas las opciones “por defecto”.
Una vez que se ha instalado el certificado raíz de la autoridad certificadora, se procede a la instalación del nuevo certificado de servidor. El cual se instala en el panel de control del IIS en la pestaña de seguridad de directorios como se observa en la figura 4-3. Se selecciona procesar la petición pendiente e instalar el certificado.

Una vez instalado se reinicia la consola del IIS y se vuelve a acceder a la pestaña "Seguridad de directorios", (una vez instalado el certificado se habilitarán las opciones de modificar y ver certificado) se pulsa en "Ver Certificado", si esta instalado correctamente se podrá observar lo siguiente:

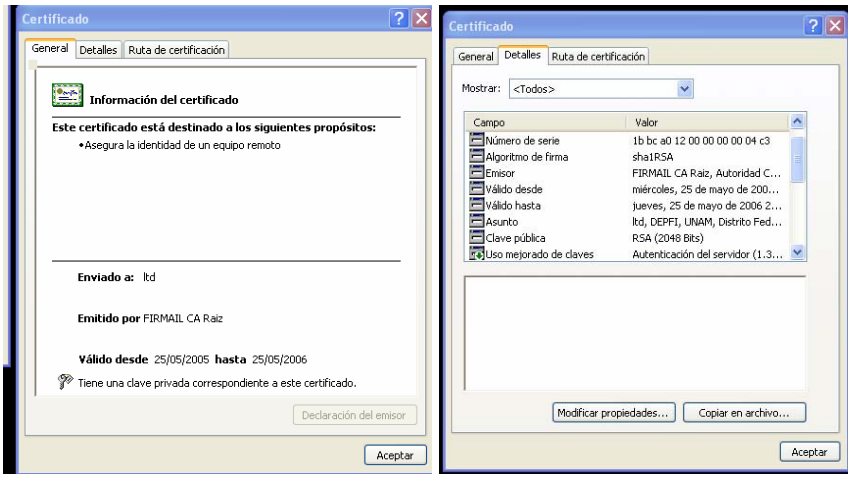


Figura 4-7. Certificado de servidor

Posteriormente se tendrán que instalar el/los certificados de las autoridades certificadoras en el almacén de certificados de la computadora "Servidor".

Se abre una consola, mediante el comando “mmc” en Menú Inicio y Ejecutar, y en agregar un nuevo complemento, se selecciona “Certificados”

Figura 4-8. Instalación del certificado en el almacén de certificados

Por ultimo hay que configurar al IIS; cada directorio y cada archivo pueden tener su propia configuración. Se abre nuevamente la pestaña "Seguridad de directorio", y se pulsa en "Modificar" y se puede seleccionar el grado de autentificación que se quiere que tenga el servidor.

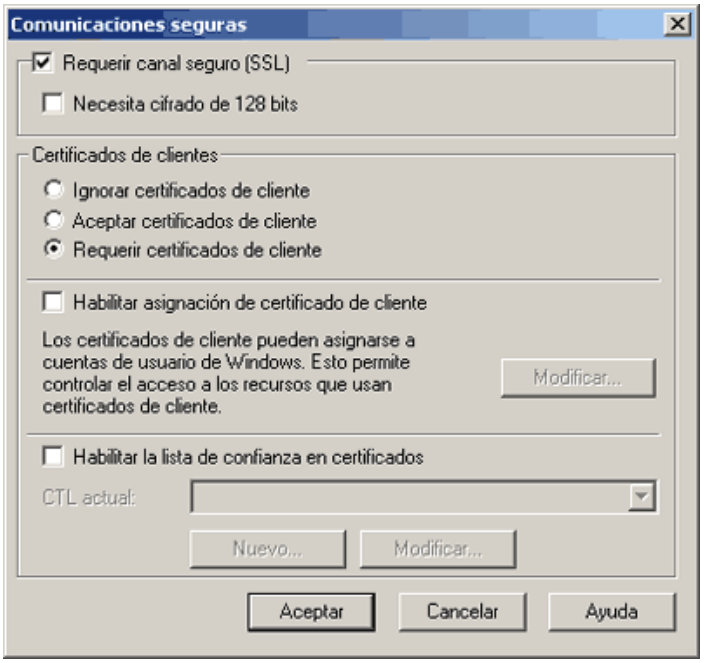


Figura 4-9. Grado de autentificación del servidor

Para configurar un directorio o un archivo en particular se hará lo mismo, pero seleccionando sus respectivos menús. Una vez terminada la configuración se puede acceder a la pagina *Web* y en ella se podrá observar un candado en la barra de estado del navegador, donde se vera el nivel de cifrado de la sesión SSL (16, 32, 64 o 128 bits).

Figura 4-10. Cifrado SSL de 128 bits.

4.4 Tecnologías y Lenguajes de Programación Empleados

Para el desarrollo del portal electrónico se utilizaron diferentes tecnologías y lenguajes de programación, es por este motivo que en esta sección se hará un análisis del código empleado, que por razones obvias no se puede describir en su totalidad ya que representa una cantidad enorme de información (mucha de ella repetitiva), sin embargo se tomarán algunos fragmentos representativos que darán una idea de la funcionalidad de cada lenguaje de programación.

Si se tienen dudas sobre alguna de las tecnologías empleadas y/o lenguajes de programación, así como, si se desea profundizar en alguno de los temas aquí señalados. Se recomienda revisar la bibliografía mostrada al final,

4.4.1 ISAPI ("Internet Server Applications Program Interface")

Interfaz de programación de aplicaciones de servidor *Internet*. Consiste en una interfaz de programación de aplicación que reside en un equipo servidor para el inicio de los servicios de "*software*", ajustados para el sistema operativo *Microsoft Windows* NT. Es una API para desarrollar extensiones para *Microsoft Internet Information Server* y otros servidores HTTP compatibles con la interfaz ISAPI [32].

4.4.1.1 Ventajas de las extensiones de servidor ISAPI

Los usuarios pueden rellenar formularios y hacer clic en un botón "Enviar" para transmitir datos a un servidor *Web* e invocar la aplicación ISA, la cual es capaz de procesar la información para proporcionar contenido personalizado, o almacenarlo en una base de datos. Las extensiones de servidor *Web* pueden usar la información de una base de datos para generar páginas *Web* dinámicamente y después enviarlas a los equipos cliente para su presentación. Su aplicación puede añadir funcionalidad personalizada y proporcionar datos al cliente mediante HTTP y HTML.

Tanto las extensiones como los filtros de servidor se ejecutan en el espacio de procesos del servidor *Web*, proporcionando una forma eficiente de extender la capacidad del servidor.

4.4.2 ASP ("Active Server Pages")

ASP proporciona un método eficiente y sencillo para crear sitios *Web* con páginas dinámicas y acceso a bases de datos. Para que un usuario realice una petición de páginas *Web*, deberá proporcionar en su explorador una dirección que indique un archivo con extensión ".asp".

Cuando se trabaja con IIS y *Active Server Pages*, el servidor de *Web* analiza las peticiones de páginas que recibe. Si se encuentra con una solicitud de una página con extensión ".asp" en lugar de ".htm", entonces se apoya en la aplicación ISAPI que sirve de soporte de ejecución de las páginas ASP.

La aplicación ISAPI de ASP reconoce las líneas HTML, de las instrucciones que dan la funcionalidad dinámica a las páginas activas. Cuando determina el lenguaje en el que se encuentran los programas escriturados ("*scripts*"), da paso al motor de ejecución de "*scripts*" adecuado (*JavaScript, VisualBasic Script*, etc.). Los motores de ejecución de *scripts* se encargan de realizar el análisis sintáctico y la compilación de las instrucciones ejecutables. Existe una memoria caché de páginas recientemente procesadas que permite aumentar las prestaciones de ASP, evitando repetir los procesos de separación de instrucciones, análisis sintáctico y compilación de las páginas más utilizadas.

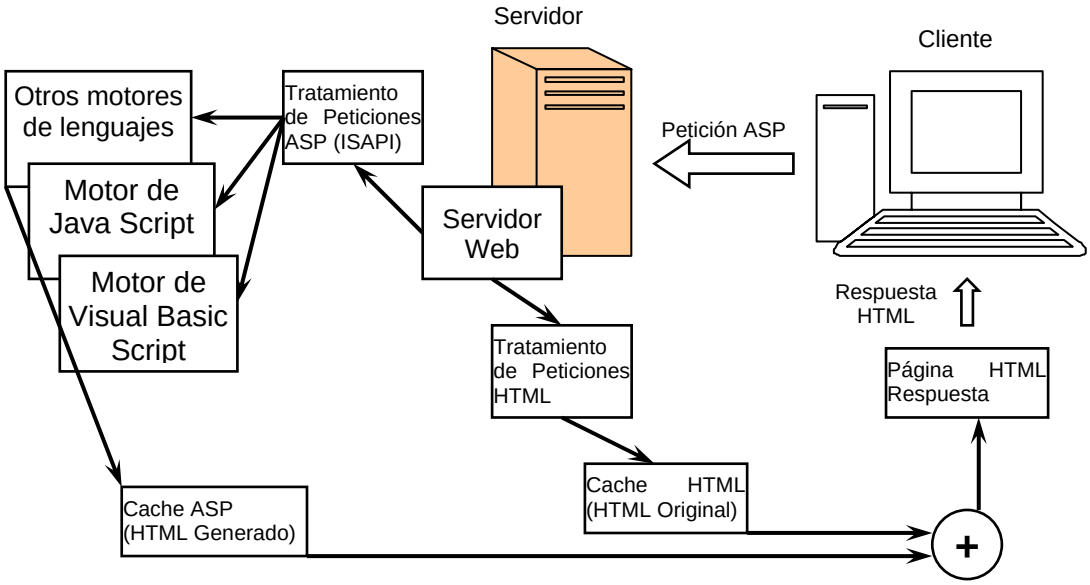


Figura 4-11. Funcionamiento General de ASP

Una vez resueltas las fases anteriores, se procede a ejecutar las instrucciones. Los motores de ejecución de *scripts* a menudo se encuentran con objetos *ActiveX* exteriores con los que tienen que interactuar. Un ejemplo muy importante de esta situación se centra en el acceso a bases de datos a través de ADO (*ActiveX Data Objects*), basados en tecnología COM (*Component Object Model*).

El usuario recibe como respuesta un archivo ".htm", que se ha formado uniendo las instrucciones HTML originales de la página ".asp" con las instrucciones HTML que se han generado tras la ejecución de los *scripts*. Para más información, se puede consultar [4].

4.4.2.1 Como conectarse a una base de datos a partir de los ODBC, utilizando los *scripts* de ASP.

El siguiente paso, una vez instalado el servidor es crear los vínculos con las bases de datos. La utilización de páginas dinámicas está muy frecuentemente asociada con el empleo de bases de datos.

Una base de datos es sencillamente un conjunto de tablas en las que se almacenan distintos registros, en este caso artículos de una tienda virtual (música, clientes, etc). Estos registros son catalogados en función de distintos parámetros que los caracterizan y que presentan una utilidad a la hora de clasificarlos. Así, los artículos de la tienda virtual podrán catalogarse a partir de distintos campos como puede ser un número de referencia, nombre del artículo, descripción, precio, proveedor.

Las bases de datos son construidas sirviéndose de aplicaciones tales como *Microsoft Access* o MySQL. El objeto aquí no es explicar la forma de explotarlas sino cómo establecer una conexión entre la base de datos, almacenada en cualquier lugar del disco duro y la página *Web* alojada también en cualquier parte y reconocida por el servidor personal a partir del directorio virtual.

Para crear este vínculo, se hace uso de los conectores ODBC (*Open Data Base Connectivity*) los cuales establecen el enlace con la base de datos.

El primer paso para crear esta conexión es ir al panel de control, a herramientas administrativas y abrir el icono ODBC. Dentro de él, se podrá crear un DSN (*Data Source Name*) de tipo sistema o usuario. Para ello se selecciona la solapa correspondiente (DSN sistema, o DSN usuario) y "Añadir". A continuación se tienen que seleccionar los controladores de la aplicación que se ha utilizado para crear la base de datos, el nombre que se le quiere asignar (aquel que se empleo en los *scripts*) y el camino para encontrarla en el disco duro.

Esta DSN permite definir la base de datos que será examinada sin necesidad de pasar por la aplicación que se haya utilizado para construirla en este caso *Access*, es decir, con simples llamadas y órdenes desde los archivos ASP se podrán obtener los datos que se busquen sin necesidad de ejecutar *Access*, MySQL, etc. los cuales, no tendrán por qué encontrarse en el servidor donde se trabaje.

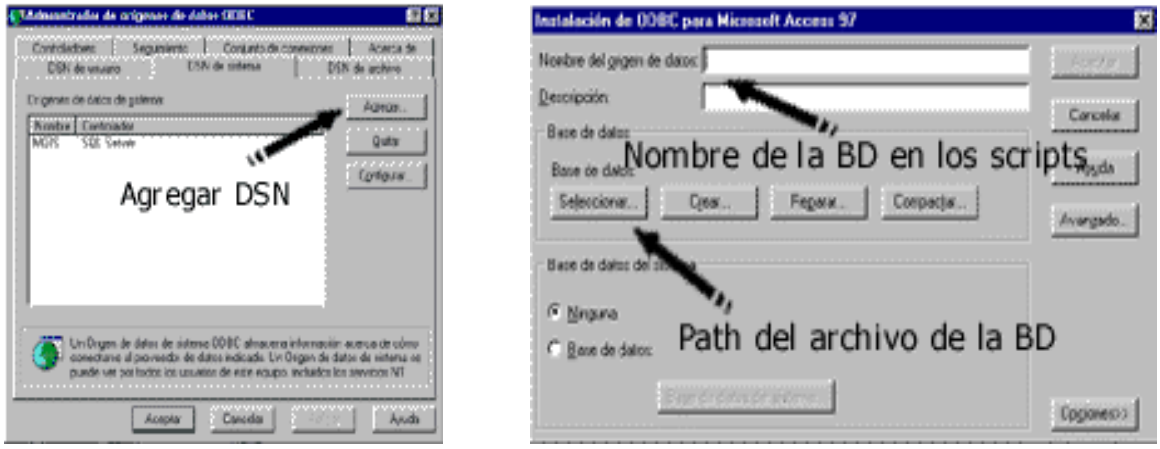


Figura 4-12. Conexión a base de datos mediante ASP

4.4.2.2 Análisis de código de ASP

Mientras que la definición de *JavaScript* se establece mediante las etiquetas <script language="JavaScript">, para ASP debe usarse la etiqueta:

```
<%@ language=jscript %>
```

En general, el lenguaje ASP entiende todas sus instrucciones siempre que estén dentro de una etiqueta cuya forma genérica es: <% --código-- %>.

Aunque ASP hace uso de los recursos del servidor para la generación de páginas HTML (lo que aumenta el tráfico del lado del servidor), tiene como puntos a su favor el ser un lenguaje de fácil aprendizaje y de relativa sencillez en cuanto a su puesta en operación.

Para el caso del portal Web desarrollado en esta tesis, se utilizó tanto para el manejo de bases de datos como para el manejo y administración de "*cookies*".

A) Bases de Datos en ASP

Para poder utilizar bases de datos a través de ASP, es necesario dar de alta la librería:

```
<!-- #INCLUDE File="ADOJAVAS.inc" -->
```

La cual contiene una gran cantidad de valores constantes que son necesarios al trabajar con bases de datos.

Un ejemplo simple del manejo de bases de datos con ASP se muestra a continuación:

```
<%
Ob_Conector = new ActiveXObject("ADODB.Connection")
// Se importa un Objeto ActiveX que establecerá la conexión con el ODBC.
Ob_RS = new ActiveXObject("ADODB.RecordSet")
// Se crea un nuevo Objeto ActiveX que almacenará los resultados.
Ob_Conector.Open("BD_Canciones")
// Se abre la conexión con el ODBC
Ob_RS.Open("Canciones", Ob_Conector, adOpenStatic, adCmdTable)
//Se indica de qué tabla se obtendrán los datos y la manera en la que estos serán obtenidos y actualizados.
Ob_RS.Filter = "IdProducto='" + Request.Form("CurrentSelec") + "'"
//Criterios de filtrado e instrucciones SQL
%>
<% while (!Ob_RS.Eof) { %>
<!—Aquí se colocaría el código que se desea extraer de la base de datos à
<%
Ob_RS.MoveNext()
}
Ob_RS.Close()
Ob_Conector.Close() %>
// Se cierra la conexión y el listado de resultados.
```

B) “*Cookies*”

Por su parte, se llaman “*cookies*” a pequeños almacenes de información (generalmente archivos tipo texto) que almacenan, administran, crean y borran los navegadores de *Internet* y que permiten guardar información referente al usuario, sus gustos, preferencias, las páginas que ha visitado, etc.

La forma mediante la cual se asigna o se extrae la información de una “*cookie*”, es mediante las instrucciones “*Response*” y “*Request*” respectivamente, siendo su sintaxis genérica la siguiente:

```
[Response/Request].Cookies(cookie)[(clave)|.atributo]=valor;
```

Y la manera de eliminar una “*cookie*” es mediante la instrucción:

```
Response.Cookies(cookie).Expires=”01/01/1980”;
```

Donde la fecha de expiración ya ha pasado.

4.4.3 JavaScript

JavaScript es un lenguaje de alto nivel, basado en objetos, diseñado para permitir a los programadores *Web* la generación de documentos HTML interactivos de un modo sencillo. Ofrece las características básicas de un lenguaje orientado a objetos sin las complejas realizaciones que acompañan a otros lenguajes como *Java* y C++. No permite la definición de clases ni la utilización de herencia.

El vocabulario de *JavaScript*, relativamente pequeño, es fácil de comprender y da un amplio número de posibilidades, antes no disponibles. *JavaScript* proporciona un conjunto de herramientas compactas propias que realizan las interacciones entre los usuarios y las páginas HTML. Estas herramientas permiten responder a las pulsaciones del ratón, a las entradas de los formularios, a la navegación de la página y a otros eventos.

Las respuestas a las acciones de los usuarios pueden ser invocadas sin necesidad de realizar transmisiones por la red. Esta es la mayor ventaja de *JavaScript* respecto a otras soluciones como ASP o CGI ("*Common Gateway Interface*"): las interacciones del usuario al ser procesadas en la computadora del propio usuario evitan la sobrecarga de tráfico en *Internet*. Con ASP o CGI, las interacciones con el usuario deben ser procesadas en el equipo servidor y, por lo tanto, transmitidas por la red.

Como la mayoría de los lenguajes de "*script*", *JavaScript* es interpretado en tiempo de ejecución por el navegador antes de que se realice. La desventaja de los lenguajes interpretados es el tiempo que se tarda en ejecutar el código, porque el navegador compila las instrucciones antes de ejecutarlas. Sin embargo, la ventaja es que son mucho más fáciles de utilizar.

Los programas *JavaScript* se insertan en las páginas HTML: si el navegador es compatible con *JavaScript* interpretará el código y lo ejecutará. Por tanto, su ejecución depende de la capacidad que tenga el navegador para interpretar el código *JavaScript* [9].

4.4.3.1 Análisis de código de *JavaScript*

JavaScript está basado en una jerarquía de objetos predefinidos que se asumen constantes en todos los navegadores para *Internet*. Además, *JavaScript* es muy útil cuando se desean controlar eventos dirigidos por acciones del usuario, o cuando se desean realizar operaciones sencillas en las que no es necesario que la página *Web* se tenga que volver a comunicar con el Servidor.

Se recomienda dirigirse a la bibliografía para entender con mayor detalle la jerarquía de objetos de *JavaScript* y los procedimientos necesarios para invocar los eventos; sin embargo, para comprender los tres ejemplos que se dan a continuación, se proporciona la gráfica siguiente:

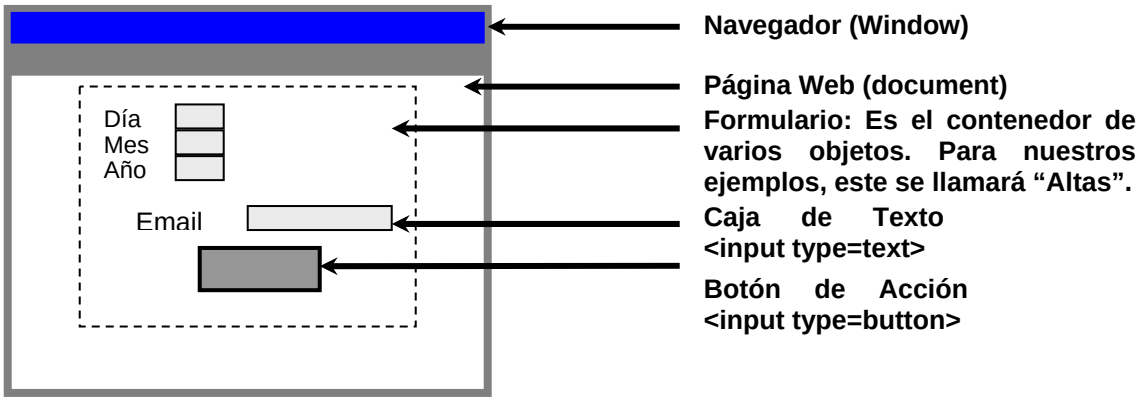


Figura 4-13. Ejemplo de una Página Web

a) Operaciones con objetos del navegador. Una de las tareas más sencillas que puede realizar un código *JavaScript* es la modificación y/o lectura de las propiedades de un objeto.

```
function ponFecha()
{
        var vDia=document.Altas.Dia.value;
        var vMes=document.Altas.Mes.value;
        var vAño=document.Altas.Año.value;

        document.Altas.Fecha.value=vDia + "/" + vMes + "/" + vAño;
}
```

La función mostrada aquí, toma los valores de tres cajas de texto contenidas dentro de un formulario llamado "Altas". Con estos valores forma una sola cadena concatenando las tres cadenas de texto e intercalando diagonales entre valores. Por último, la cadena final es asignada a otra caja de texto del mismo formulario.

b) Verificación de Formularios. Otra de las actividades en las que constantemente se recurre a los *scripts* de *JavaScript* es en la verificación de los campos de formularios.

```
function VerificaCampos()          {
                if(document.Altas.Email.value=="")
{
window.alert("No se introdujo la dirección de e-mail");
}
else
{
window.alert("¡¡¡Gracias!!!");
        Alta(document.Altas.Email.value);
                }
}
```

En este ejemplo se ve que al invocarse la función *VerificaCampos*( ), se busca la casilla de E-mail y se verifica su valor. Si la casilla está en blanco, el *script* envía un mensaje de

error para solicitar que el usuario corrija la falta. En el caso de que exista el valor, lo agradece y lo envía como parámetro a otra función, que es la encargada de dar de alta la dirección de correo electrónico.

c) Eventos. Por último, el manejo de eventos es de las funciones más sencillas pero al mismo tiempo de las más potentes a la hora de trabajar con páginas HTML.

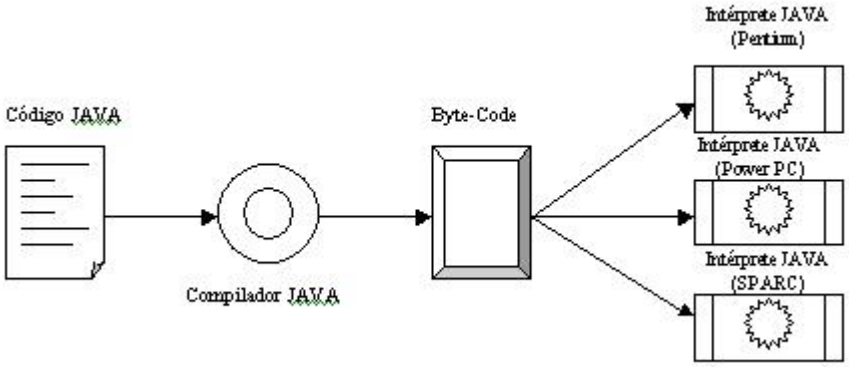
```
<input name="Enviar" value="Enviar" type=button OnClick=VerificaCampos()>
```

En este ejemplo se puede ver que se ha incorporado dentro de la página *Web* una etiqueta de entrada tipo botón. Este botón tiene un nombre y un valor, pero al realizarse la acción de pulsarlo (*OnClick*), se hace un llamado a la función *VerificaCampos*( ), vista con anterioridad.

4.4.4 Java

Java es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems*. Fundamentado en C++, *Java* se diseño para ser: pequeño, sencillo y portátil, a través de plataformas y sistemas operativos, tanto en nivel de código fuente como binario, lo que significa que los programas *Java* pueden ejecutarse en cualquier computadora (una PC con *Windows*, un computador basado en el sistema operativo *Unix*, etc.); es decir, el programador solo debe escribir el programa una vez, y lo puede ejecutar en cualquier computadora.

Los programas en *Java* generalmente son compilados en un lenguaje intermedio llamado *bytecode*, para luego ser interpretados por la *Java Virtual Machine* (JVM). Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje, permitiendo que se pueda "escribir el programa solo una vez, y correrlo en cualquier lado".



4-14. Compilador Java

Las características propias del lenguaje *Java* hacen que además de poder desarrollar aplicaciones que se ejecutan en el intérprete local, se puedan desarrollar módulos descargables a través de una página *Web* y ejecutables en la JVM del navegador. Estos módulos reciben el nombre de *Applets*.

- Aplicaciones
  - Escribir el programa fuente en cualquier editor y guardarlo con extensión .java

  - Compilar el archivo fuente mediante: javac miPrograma.java. Esto genera el archivo .class
  - Ejecutarlo (interpretar los *byte-code*) : java miPrograma
- *Applets*
  - Escribir el programa fuente en cualquier editor y guardarlo con extensión .java
  - Compilar el archivo fuente mediante: javac miProgramaApplet.java
  - Escribir la pagina *web* que contendra al *applet* y guardar el código con extension .html

4.4.4.1 JDK ("Java Developer Kit")

"*Java Development Kit*"(JDK) Herramienta de Desarrollo de *Java*,"*Standard Development Kit*" (SDK) y "*Java* 2 *Standard Edition*" (J2SE) son nombres para el mismo componente y se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en *Java*.

Para desarrollar programas en *Java* es suficiente con instalar el paquete JDK de *Sun*, que es de libre distribución. En el portal *Web* de *Sun* se puede encontrar toda clase de información relacionada con *Java* [40]: ejemplos de programas escritos en *Java*, tutoriales, documentación, errores conocidos (*bugs*) y su solución; entre otros documentos. La última versión disponible hasta la fecha es la JDK 5.

El JDK contiene una serie de clases, el compilador de *java* (javac), el visor de *applets* (*appletviewer*), el intérprete *java* (java), y otros programas con propósitos específicos.

Javac: Es el comando compilador de Java. Su sintaxis es:

Javac ejemplo.Java

La entrada de este comando ha de ser necesariamente un archivo que contenga código escrito en lenguaje *Java* y con extensión .Java. El comando creará un archivo *.class por cada clase que contenga el archivo Java. Los archivos *.class contienen código *bytecode*, el código que es interpretado por la máquina virtual *Java*.

Java: Es el intérprete de *Java*. Permite ejecutar aplicaciones que previamente hayan sido compiladas y transformadas en archivos *.class. Su sintaxis es:

Java ejemplo

No es necesario aquí suministrar la extensión del archivo, ya que siempre ha de ser un archivo *.class.

Appletviewer: Se trata de un comando que verifica el comportamiento de un *applet*. La entrada del comando ha de ser una página *Web* que contenga una referencia al *applet* que se desea probar. Su sintaxis es:

Appletviewer mipagina.html

El comando ignora todo el contenido de la página *Web* que no sean *applets* y se limita a ejecutarlos. Un ejemplo de página *web* "mínima" para poder probar un *applet* llamado miapplet.class sería:

```
<HTML>
<TITLE>My Applet </TITLE>
<BODY>
<APPLET CODE="miapplet.class" WIDTH=180 HEIGHT=180>
</APPLET>
</BODY>
</HTML>
```

Además de estos programas JDK cuenta con un buen número de librerías para propósitos particulares, para la elaboración de esta tesis se utilizó una librería de Java llamada "Cryptix" para el cifrado y decodificación de los archivos de música, así como para la aplicación cliente del usuario, la cual será analizada más adelante.

4.4.4.2 La maquina virtual de Java (JVM)

Tal y como se ha mencionado, la existencia de distintos tipos de procesadores y ordenadores llevó a la conclusión de que era muy importante procurarse un *software* que no dependiera del tipo de procesador utilizado. Se planteó la necesidad de conseguir un código capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado no debería ser necesaria ninguna modificación por el hecho de cambiar de procesador, o de ejecutarlo en otra máquina. La clave consistió en desarrollar un código "neutro" el cual estuviera preparado para ser ejecutado sobre una "*máquina hipotética o virtual*", denominada *Java Virtual Machine* (JVM). Es esta JVM quien interpreta este código neutro convirtiéndolo a código particular de la CPU utilizada.

La JVM es el intérprete de *Java*. Ejecuta los "*bytecodes*" (archivos compilados con extensión ****.class***) creados por el compilador de *Java* (javac.exe). Tiene numerosas opciones entre las que destaca la posibilidad de utilizar el denominado ***JIT*** (*Just-In-Time Compiler*), que puede mejorar entre 10 y 20 veces la velocidad de ejecución de un programa.

4.4.4.3 Implementación de Java en el sitio de comercio electrónico

Para la realización de este trabajo de tesis se utilizo *Java*, para la implementación del sitio de comercio electrónico desde la utilización de *applets* para la elaboración de la página *Web* que se encargara del cifrado de datos, en este caso de la música encriptada, así como para la creación del reproductor de la música encriptada para ello se empezó a recurrir a las herramientas y arquitectura criptográficas ya desarrolladas por el lenguaje *Java*, como son las librerías de encriptación del JCE de Cryptix. Esto era necesario para crear el ejemplo de la concepción básica del "*CryptoMusicMaker*", donde una página *Web* es usada para cifrar archivos con extensión "***.txt***" utilizando alguno de los algoritmos de cifrado siguientes: DES, Blowfish o AES.

En las secciones siguientes se explicara mas en detalle el usó de la librería de *Java* llamada "Cryptix" para el cifrado y descifrado de los archivos de música, así como para la aplicación cliente del usuario, y también se mostraran las aplicaciones creadas con *Java*

como son el "*CryptoMusicMaker*", el *"Crypto Player"* y el uso de funciones resumen (hash) en la implementación del sitio *Web*.

4.4.5 HTML y D-HTML ("Hipertext Markup Language" y "Dynamic HTML")

HTML es el código estándar para la creación de páginas *Web*. Recientemente, sin embargo, ha aparecido una serie nueva de etiquetas y propiedades para las etiquetas anteriormente definidas que permiten una mayor interacción y una mejor interfase de usuario [4].

4.4.5.1 HTML

HTML o "lenguaje de formato de documentos de hipertexto", es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas *Web*. Gracias a *Internet* y a los navegadores, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en el navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del computador. Toda etiqueta se identifica porque está encerrada entre los signos "menor que" y "mayor que" (<>), y algunas tienen atributos que pueden tomar algún valor.

Las etiquetas básicas de HTML son:

- `<HTML>`: Es la etiqueta que define el inicio del documento html, le indica al navegador que todo lo que viene a continuación debe tratarlo como una serie de códigos html.

- `<HEAD>`: Define la cabecera del documento html, esta cabecera suele contener información sobre el documento que no se muestra directamente en el navegador. Como por ejemplo el título de la ventana de su navegador. Dentro de la cabecera `<HEAD>` se puede encontrar:
    - `<TITLE>`: Define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana
    - `<LINK>`: Se utiliza para definir algunas características avanzadas, como por ejemplo las *stylesheets* (hojas de estilo) usadas para el diseño de la página, ejemplo :`<link rel="stylesheet" href="/style.css" type="text/css">`

- `<BODY>`: Define el contenido principal o cuerpo del documento, esta es la parte del documento html que se muestra en el navegador, dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo `<BODY>` se pueden encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:

    - `<H1>, <H2>,... <H6>`: encabezados o títulos del documento en diferentes tamaños de fuente
    - `<P>`: párrafo nuevo

-
: salto de línea forzado
- <TABLE>: comienzo de una tabla (las filas se identifican con <TR> y las celdas dentro de las filas con <TD>)
- <A>: indica la existencia de un hipervínculo o enlace, dentro o fuera de la página *Web*. Debe definirse el parámetro de pasada por medio del atributo *href* (ejemplo: `<a href="www.criptomusic.com">Criptomusic</a>` se representa como *Criptomusic*)
- <DIV>: comienzo de un área especial en la página
- <IMG>: indica la existencia de una imagen para mostrarse en el navegador

Ejemplo de código con etiquetas básicas en HTML.

```
html>
  <head>
    <title>Ejemplo</title>
  </head>
    <body>
        <p>ejemplo</p>
    </body>
</html>
```

4.4.5.2 DHTML

Anteriormente, cuando una organización creaba un portal *Web*, debía asegurarse que todas las ligas apuntaran correctamente, que los colores de letras fondos o imágenes (por ejemplo, el logotipo de la empresa) correspondieran y estuvieran en el sitio adecuado. El problema de seguir este esquema radica en que, si se deseaba hacer un cambio en todas las páginas *Web*, dicho cambio debía reproducirse tantas veces como páginas hubiera en el portal.

La incorporación del HTML Dinámico permite ahorrar trabajo de administración y actualización al utilizar hojas de estilo (CSS); así por ejemplo, si se tiene el siguiente código dentro de una página Web:

<style>@import URL("Estilo_Form.css");</style>

y al mismo tiempo se cuenta con un archivo llamado "*Estilo_Form.css*", cuyo contenido es el que se muestra a continuación:

Estilo_Form.css

```
body {color:'#000000'; Font-family:arial; text-align:justify;}
h1 {color:'#000000'; text-align:center; font-size:18;
font-family:Arial; font-style:italic; font-weight:bold}
h2 {color:'#FF0000'; text-align:justify; font-size:13; font-style:Arial}
h3 {color:'#FF5500'; text-align:justify; font-size:13; font-style:Arial}
```

Al incorporar esta etiqueta dentro de la cabecera de las páginas HTML, se estaría forzando a que todas las páginas tengan un color de fondo negro y tengan un tipo de letra "por defecto" *arial* con alineación justificada (etiqueta <body>).

De igual modo, si dentro de la página *Web* se etiqueta un párrafo con <h1>, se indicaría que dicho párrafo tendría un color de letra negro, su alineación estaría centrada, el tamaño y fuente sería 18 y *arial* respectivamente, tendría activa la propiedad de remarcado. Y así sucesivamente.

Como puede verse, la utilización de hojas de estilo ahorra tiempo cuando se desea homogeneizar la presentación de las páginas estableciendo un formato predefinido. De igual forma, cuando se desea cambiar el aspecto de todas o algunas de las páginas *web*, solamente será necesario modificar el archivo "*.css".

4.5 Componentes de Seguridad Implementados en el Sitio de Comercio Electrónico

A continuación se describen los diferentes mecanismos de seguridad utilizados para simular el portal de comercio electrónico en *Internet*, tratando de cubrir los aspectos más relevantes de seguridad con los que se cuenta en la actualidad.

4.5.1 Componentes No Criptográficos

Una vez conocidas las vulnerabilidades y ataques a las que esta expuesto un sistema es necesario conocer los recursos disponibles para protegerlo. Mientras algunas técnicas son evidentes (seguridad física) otros modelos no lo son tanto, e incluso unas pueden ocasionar una sensación de falsa seguridad.

La criptografía es un componente fundamental de una solución de seguridad informática para sistemas distribuidos; pero debe trabajar en conjunto con otros componentes no-criptográficos necesarios para integrar la solución acorde a la realidad de un sistema seguro; esquemas de redundancia energética y funcional, "*firewalls*", seguridad física de establecimientos, computadoras y sistemas operativos seguros, etcétera (tal y como se analizo en el capitulo anterior).

4.5.1.1 Programa antivirus

Un antivirus es una gran base de datos con la huella digital de todos los virus conocidos para identificarlos y también con las pautas que más contienen los virus. Los fabricantes de antivirus avanzan tecnológicamente casi en la misma medida que lo hacen los creadores de virus. Esto sirve para combatirlos, aunque no para prevenir la creación e infección de otros nuevos.

Actualmente existen técnicas, conocidas como heurísticas, que brindan una forma de "adelantarse" a los nuevos virus. Con esta técnica el antivirus es capaz de analizar archivos y documentos y detectar situaciones sospechosas.

4.5.1.2 "*Firewalls*"

Dado que la tesis debe abarcar la transmisión de información en redes de datos, éstas posibilitan muchos ataques internos y externos. Por ello, el uso de los "*firewalls*" directamente evita el acceso de atacantes a los recursos informáticos así como la realización de ataques. Pero por las características propias de su funcionamiento, no son la absoluta defensa para todo tipo de ataque por red, ya que los *firewalls* no son sistemas inteligentes, ellos actúan de acuerdo a parámetros introducidos por su diseñador, por ende si un paquete de información no se encuentra dentro de estos parámetros de peligro simplemente lo deja pasar

4.5.1.3 Sistemas Detectores de Intrusos

Por la misma razón, los sistemas detectores de intrusos localizan, alertan y, si están integrados con los *firewalls*, frenan a los atacantes cuando quieran acceder, diagnosticar o atacar la infraestructura informática de la empresa.

Se recomienda leer [24]

4.5.2 Componentes Criptográficos

En esta sección se mostraran los elementos criptográficos necesarios para el modelo propuesto.

4.5.2.1 Librería Cryptix

Cryptix es un esfuerzo internacional cuyo objetivo es producir una serie de librerías y código fuente criptográfico abiertos y robustos. Este producto es de acceso libre y puede ser utilizado tanto en implementaciones comerciales como no comerciales. Actualmente esta siendo usado por desarrolladores de todo el mundo. Su código fuente se basa en el lenguaje de programación *Java*.

De acuerdo a la página oficial de *Java* en *SUN*, la Extensión Criptográfica de Java (JCE) provee un marco de trabajo e implementaciones para cifrado, generación de llaves, convenciones para el uso de llaves, y algoritmos para la autentificación de mensajes. Soporta además cifrado simétrico, asimétrico, de bloque y de flujo. Esta extensión también da soporte para la creación de "flujos seguros" y objetos firmados.

Sin embargo, la página oficial de *Java* no provee a los desarrolladores internacionales de la implementación del JCE, esto debido a que las políticas de exportación de los Estados Unidos impiden la salida de material criptográfico clasificado fuera de sus fronteras. Cryptix JCE comenzó a desarrollarse para resolver este problema. Cryptix JCE es un conjunto de algoritmos y programas que se ajustan al estándar general esbozado por el API del JCE 1.2 oficial publicado por *SUN*. Cryptix JCE espera ser compatible 100% con la implementación de *SUN* y por supuesto, está disponible a nivel internacional de forma libre.

Cryptix tiene a su disposición los siguientes algoritmos de cifrado:

Blowfish	CAST5	DES

IDEA	MARS	RC2
RC4	RC6	Rijndael
Serpent	SKIPJACK	Square
TripleDES	Twofish	

Tabla 4-3. Algoritmos Criptograficos soportados por Cryptix

Soporta además la convención de intercambio de llaves propuesta por Diffie-Hellman

Modos de Trabajo		
CBC	CFB-(con tamaños e bloque de 8, 16, 24, ..., bytes)	ECB
OFB (con distintos tamaños de bloque)		openpgpCFB

Funciones Resumen		
MD2	MD4	MD5
RIPEMD-128	RIPEMD-160	SHA-0
SHA-1	SHA-256/384/512	Tiger

Códigos de Autentificación		
HMAC-MD2	HMAC-MD4	HMAC-MD5
HMAC-RIPEMD-128	HMAC-RIPEMD-160	HMAC-SHA-0
HMAC-SHA-1	HMAC-Tiger	

Firmas Digitales		
RawDSA	RSASSA-PKCS1	RSASSA-PSS

Cifradores Asimétricos: RSA / PKCS#1

4.5.2.2 Uso de Certificados

Para la creación de certificados se requirió del uso de mecanismos criptográficos; y para el desarrollo del simulador se tuvieron que crear dos tipos de certificados: uno para la creación de *applets* y otro para el establecimiento de un canal de comunicación seguro.

Creación de *Applets*

Para poder ejecutar los *applets* utilizados en la aplicación del "*CriptoMusicMaker*" es necesario tener que crear un certificado de autentificación ya que los *applets* están diseñados para efectuar operaciones que "pueden ser consideradas como peligrosas", tal como escribir en el disco duro. Por esta razón, si cualquier usuario ve una advertencia al navegar por *Internet*, deberá estar seguro de que, quien escribió un "*applet*" con estas características, es un programador de confianza, pues de otra forma, su información puede verse comprometida.

Cuando los sitios en *Internet* cuentan con este tipo de certificador, pero dichos certificados han sido emitidos por entidades comerciales de confianza (tales como Verisign o e-Trust), el certificado se instala en el navegador del cliente de forma transparente. Esto se debe a que, al ser entidades de confianza han establecido de antemano acuerdos y licencias con los productores de "*software*" (*Microsoft, Netscape*, etc.), de forma tal que los certificados que estas empresas emiten están listos para aplicaciones de comercio electrónico.

Por tal motivo al no contar con un certificado emitido por entidades de confianza, debido a que el costo de los certificados digitales no es bajo; para esta tesis se auto crearon certificados con los cuales se trabajó durante la etapa de desarrollo (el JDK de *Microsoft* así como el de *Sun* tienen herramientas propias para crear este tipo de certificados "no legales").

Firma de applets para Internet Explorer

Resulta necesario tener instalado el JDK para Java 3.1 o superior. La ventaja de la creación de *applets* firmados para *Internet Explorer* es que no se necesita añadir código especial al archivo fuente del *applet*, ni realizar llamadas a funciones criptográficas. Para firmar cualquier *applet* no se requiere modificarlo en absoluto.

Una vez que se dispone de las herramientas necesarias, se puede proceder a firmar *applets* siguiendo los siguientes pasos:

Paso 1: Crear un certificado para firmar *applets*
En el caso de *Microsoft* se necesita un certificado adecuado para su modelo de *Authenticode*. Existen muchas autoridades de certificación que distribuyen tales certificados, como *VeriSign*. Para uso personal se debe solicitar un certificado de clase 2, mientras que para uso comercial se necesita uno de clase 3.

En el directorio \bin\PackSign (creado cuando se instala el JDK) se encuentran las herramientas en línea de comandos que crearán un certificado de prueba. Para la creación del certificado, se hará lo siguiente:

makecert /sv "CryptoMaker.pvk" /n "CN=CryptoMaker Certificate" CryptoMaker.cer

Donde:

- `CryptoMaker.pvk` es el nombre del archivo de la llave privada generada
- `CryptoMaker.cer` es el archivo del certificado generado
- `CryptoMaker es el nombre del certificado`

Durante esta fase se requiere de una contraseña en dos ocasiones. Tiene que ser la misma contraseña en ambos casos y va a ser requerida posteriormente

Sin embargo, este certificado no sirve para firmar código. En su lugar se necesita un Certificado de Productor de *Software* (*Software Publisher Certificate*, SPC), el cual debe obtenerse de una AC a la que se envía este certificado recién generado para que lo autentique y lo firme. En su defecto, y sólo con fines de prueba, existe otra herramienta

del JDK que permite transformar el certificado anterior en uno válido, de la siguiente forma:

- Cert2spc CryptoMaker.cer CryptoMaker.spc

Ahora ya se puede utilizar este certificado, CryptoMaker.spc, para firmar código, con prestaciones similares a las que poseería si se hubiera obtenido de una AC.

Paso 2: Crear el CAB

La firma basada en *Authenticode* funciona con archivos armario (Cabinet, CAB). Los archivos CAB constituyen simplemente una manera de compactar varios archivos en uno solo en un formato que pueda entender IE. Para archivar varios archivos se utiliza la herramienta cabarc. Por ejemplo, para comprimir los archivos fich1.class y fich2.class en uno solo se utilizaría:

- cabarc n encriptadores.cab encriptadores1.class encriptadores2.class

Donde la opción n indica que se quiere crear un nuevo archivo.

Paso 3: Firmar el CAB

El JDK incorpora la herramienta *signcode* para firmar código. En este paso conviene decidir qué nivel de seguridad se le desea asignar, ya que en función de dicho nivel y de la configuración de las zonas de seguridad en el navegador del usuario, el resultado será que se presente un determinado número de ventanas pidiendo al usuario confirmación antes de permitir que el *applet* “escape” de los confines del recinto de seguridad. El comando para firmar el código es:

- signcode –spc CryptoMaker.spc –k CryptoMaker encriptadores.cab

Para confirmar que todo el proceso se ha realizado correctamente, se puede ejecutar el comando: chkjava CryptoMaker.cab

“Por defecto”, *signcode* requiere permisos totales, como se ha podido comprobar al ejecutar el comando chkjava. Si se desea, se puede añadir información de permisos de *Java* a la firma para controlar más finamente el tipo de recursos a los que tendrá acceso el *applet*, usando la opción -j JavaSign.dll de *signcode*.

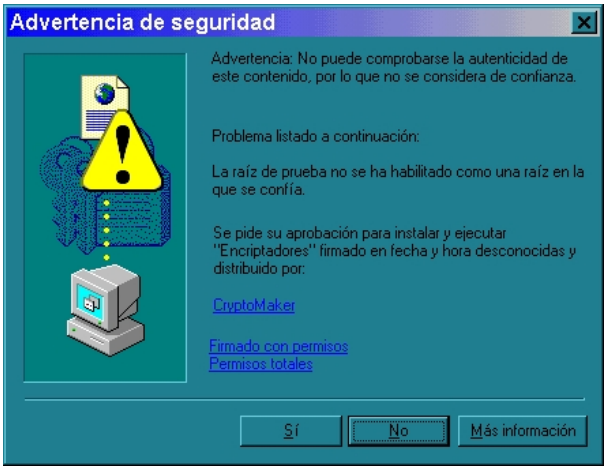
Paso 4: Incrustar el CAB en la página *web*

Ya sólo resta incluir las etiquetas correctas para que se visualice el *applet* sin problemas, lo cual exige variar ligeramente el formato de la etiqueta <applet> convencional:

```
<applet code="Encriptadores.class"> <param name="cabbase"
value="Encriptadores.cab"> </applet>
```

Es importante eliminar el archivo Encriptadores.class original del directorio, ya que en caso contrario el navegador lo cargaría en vez del firmado. El *applet* ya está listo para ejecutarse en *Internet Explorer*. En adelante, cuando el visitante a la *Web* cargue la página donde se aloja el *applet* firmado, le aparecerá una advertencia de seguridad solicitándole su aprobación para instalarlo y ejecutarlo, ver figura siguiente. Si el visitante

acepta, el *applet* se ejecutará para acceder sin restricciones a todos aquellos recursos que se solicitaron en el archivo *.ini de permisos, creado mediante la herramienta PIniEdit y utilizado en el proceso de firmado. Puede examinar con más detalle los permisos solicitados, para decidir si aprobar o no su ejecución. Si el visitante deniega los permisos, entonces el *applet* no se ejecuta en absoluto.



4-15. Advertencia para un certificado de Applet

Canal Seguro

Conviene utilizar un canal de comunicación seguro el cual, con la finalidad de poder realizar la transferencia de bits en forma segura por la red pública de *Internet*, incluye los procesos de:

1. Petición y confirmación de petición de un canal de comunicación seguro.
2. Establecimiento del canal y realización de la comunicación.
3. Fin de la comunicación y cierre del canal.

La implementación del canal seguro se realizo mediante la instalación del certificado de Servidor como se vio en la sección 4.2.2.3.

4.5.2.3 “CryptoPlayer”

Realiza como acción la ejecución de “la canción”, para ello necesita:

1. Factura digital de compra.
2. La canción cifrada, de forma que dependa de la factura y la identidad del usuario.
3. Identidad del usuario.

4.5.2.4 “CryptoMusicMaker”

Crea la canción personalizada para cada Cliente Final.
Requiere:

La llave de encriptación de la Casa Musical.
La canción cifrada por la Casa Musical.
Datos del Cliente Final.

Factura del Cliente Final.

Da:

1. La canción personalizada al Cliente Final

*Importante: la canción o la factura deben llevar en "texto en claro" título, cantante, autor, etc.; es decir, los datos que pudieran ser usados para un ataque por "texto en claro" conocido NO deben de estar cifrados.

4.5.2.5 "Crypto Engine de Autorización"

Protege la "e-form" de autorización que será enviada al Banco para la transferencia de fondos de la cuenta de uno de sus cuenta-habientes a otras cuentas, así como las formas electrónicas que mandan los cuenta habientes empresariales (Tienda Virtual y Casa Musical) para realizar las operaciones mercantiles.

4.6 Procedimientos de Protección

Por último, mostramos las consideraciones y procedimientos que constituyen los protocolos para el manejo de los bienes a proteger.

4.6.1 Manejo de canciones

El Modelo General a usar entre Tienda Virtual y Casas Musicales establece la compra-venta de paquetes de canciones y canciones selectas; ambos pueden ser de carácter temporal (la llave y la canción caducan, para protegerlas en caso de que se hayan comprometido en el almacén de la Tienda Virtual) o permanente. Esto obliga a usar llaves de encriptación temporales o permanentes; "*timestamp*" (para indicar la fecha de inicio de vigencia) y tiempo de vida de las llaves temporales.

Esta situación también obliga a cada Casa Musical a dar su propio sistema encriptador "*CryptoMusicMaker*" para que sólo la Tienda Virtual que las haya comprado pueda hacer uso de la misma, además de tener la capacidad de recibir como valores de entrada llaves y datos para particularizar la canción (en un formato estándar) al cliente que se le vendió.

Este esquema implica que cada Casa Musical le da a la Tienda Virtual su "crypto engine", llamado "*CryptoMusicMaker*", llaves y canción cifrada para crear las canciones a vender al Cliente Final:

- Si la llave, la canción, o el "*CryptoMusicMaker*" son incorrectos o caducos; no se crea un archivo de salida (canción a dar al Cliente Final).
- El "*CryptoMusicMaker*" nunca deja una canción en texto en claro en algún registro.
- Las entradas al "*CryptoMusicMaker*" son:

    1. Canción cifrada.
    2. Llave de Tienda Virtual dada por la Casa Musical.
    3. Fecha del sistema.

4. Llave para cifrar la canción para el Cliente Final; y verificación que no sea una llave prohibida (que de como resultado el texto en claro).
5. Campos a anexar a la estructura de la canción (los cuales verifica el *CryptoPlayer*).

- El "*CryptoMusicMaker*" verifica la integridad, "timestamp", firma de la Casa Musical y vida útil de:
  - Llaves usadas por la Tienda Virtual.
  - Canción.

- La salida del "*CryptoMusicMaker*" es una canción cifrada que solo puede ser tocada por el Cliente Final.

En este modelo se propone que la Tienda Virtual venda canciones digitalizadas personalizadas a las características del cliente, para que él y únicamente él pueda reproducirlas. Además se utilizará el concepto del mundo común: el dueño de un bien debe tener un documento que acredita su posesión; por lo cual también será necesario utilizar una factura digital con el fin de que el cliente acredite la posesión de su(s) canción(es).

Las primeras consideraciones a usar son:

- La existencia de información única del cliente, la cual es intrínseca a él y diferente a la de otros clientes. Por lo tanto esta información única debe ser utilizada como un parámetro para "particularizar" la canción.
- El uso de dicha información única del cliente por parte de la Tienda Virtual implica que ésta también la conoce. Dicha información puede ser: la generación de una llave o clave en el momento en el que el usuario se dio de alta en la Tienda Virtual, además de los datos propios que da el usuario durante este proceso.

Para que el cliente escuche la canción debe usar un reproductor llamado "*CryptoPlayer*", el cual tiene las siguientes características:

- El Cliente Final puede instalarlo en todas "sus" máquinas. El hecho de que se use en varias máquinas implica que la información única del Cliente Final no depende del "*hardware*" o sistema operativo, por lo tanto se puede usar como información única los datos que dio cuando se dio de alta en la Tienda Virtual y la clave que ésta le haya asignado. Además el "*CryptoPlayer*" solo puede estar instalado para reconocer a un solo usuario.
- El "*CryptoPlayer*" conoce y usa la información única del Cliente Final para poder reproducir dichas canciones que el compra. Dicha información debe estar guardada en algún registro, cifrado, por lo cual no pueda ser visto por alguien ajeno al propio programa.
- Por seguridad, el "*CryptoPlayer*" debe se inicializado con un "*password*" al inicio de cada sesión.
- El "*CryptoPlayer*" usa una base de datos para almacenar las canciones que compra el usuario y las facturas correspondientes a cada canción.

Esto significa que si el Usuario Final desea escuchar una canción, debe tener los siguientes elementos para que funcione el "*CryptoPlayer*" (verifique y descifre correctamente):

- El "*password*" para iniciar la sesión del "*CryptoPlayer*".
- Los datos de usuario dados en la instalación (nombre completo, RFC, etc.), los cuales deben corresponder a los establecidos en la factura (usuario legítimo).
- La clave que se le asignó al darse de alta como cliente de la Tienda Virtual.
- La canción cifrada, la cual debe corresponder a la establecida en la factura (canción legítima) en cuanto a nombre y tamaño.
- La factura de dicha canción, con la misma relación de información y datos, tanto de la canción como del usuario (el usuario es legalmente dueño de la canción).
    - Además, el "*CryptoPlayer*" deberá verificar la integridad, la firma y el "*timestamp*", tanto en la canción(es), como en la factura.

Se puede diseñar el "*crypto engine*" para que todo el mensaje cifrado sea una combinación de los 3 elementos anteriores, y luego usar la llave extra proporcionada por la Tienda Virtual. O bien, la llave se forma solo con los 3 elementos anteriores.

Resumiendo lo anterior, la Tienda Virtual también tiene una naturaleza dual por el tipo de comercio electrónico a establecer según sus participantes:

- Al vender sus canciones a los usuarios finales, requiere que su operación sea B2C (*Business to Customer*, negocio a consumidor).
- Al comprar las canciones a las Casas Musicales, requiere que su operación sea B2B (*Business to Business*, negocio a negocio) en un grado poco desarrollado.

Concluyendo con esto, se ejemplifica un negocio en el cual se conjugan los 2 tipos de comercio electrónico más comunes: el B2B y el B2C; unidos por un ente común: la Tienda Virtual.

4.6.2 Manejo del dinero y Autorización Electrónica

Se asume que el representante legal de la Tienda Virtual y de la Casa Musical, así como el Cliente Final, se dan de alta como usuarios de sus respectivos Bancos FISICAMENTE (lo cual significa que no existen altas al Banco por medios electrónicos). Esto es con el fin de que la autentificación de los entes dueños de una cuenta, sea realizada por medio de documentos oficiales (ya que en la actualidad definen el nivel máximo de autentificación social). Esta actividad se realiza en las sucursales de cada banco, y es cuando se les asigna a cada uno el Número de Cuenta, el Número de Tarjeta asociada a dicha cuenta y su clave para descargar las herramientas electrónicas necesarias para operaciones electrónicas.

Una vez que cada ente ya es cuentahabiente del Banco, descargará las formas electrónicas y herramientas criptográficas necesarias para poder realizar sus operaciones comerciales por *Internet*. Para dicha descarga, así como para usar los servicios de banca electrónica (consulta de saldo, transferencia de fondos entre diferentes cuentas, pago de servicios domiciliados, etc.); cada cuentahabiente (particular o representante legal) primero deberá entrar al sistema y autentificarse con su clave.

Para las operaciones comerciales por *Internet*, en la actualidad el "pago a terceros con cuentas de diferentes bancos" no esta establecido; aunque esta en estudio su implementación. Por el momento como una opción a esta situación, se utiliza el pago interbancario (más indirecto y no en línea conforme a las políticas de la red SECOBAN). Para los propósitos de esta tesis, manejaremos un posible esquema de "pago a terceros con cuentas de diferentes bancos" en el cual se maneja una forma electrónica que funcionará como "autorización cifrada". Otras formas de pago se pueden ver en [8] y [26].

El Banco crea sus "e-forms" para "autorizaciones de transferencia" y "peticiones de transferencia entre cuentas", y los algoritmos criptográficos para proteger dichos documentos (algoritmos implementados como un "*crypto engine*", y pueden ser distintos para cada banco); los cuales pueden ser descargados por cada uno de sus clientes registrados.

Una "autorización de transferencia" de dinero entre diferentes cuentas, es análoga a la firma de conformidad para validar un pagaré. Este documento consta de:

- Número de cuenta emisora (comprador)
- Número de cuenta receptora (vendedor)
- Monto autorizado por ambas partes
- "*Timestamp*" del comprador
- Firma digital del comprador

Al comprar un usuario del Banco por *Internet*-WWW, debe de dar su autorización que será canalizada por el negocio cibernético. Dicha autorización es cifrada con el "*crypto engine*" del banco poseedor de la cuenta del usuario, con el fin de que este documento solo sea visible para el comprador y para el Banco; y así evitar atentados por parte del vendedor.

Para realizar la operación de compra-venta, el negocio debe canalizar la "autorización de transferencia" que le otorga el cliente como parte de la "petición de transferencia entre cuentas", la cual cifra la Tienda Virtual con el "*crypto engine*" de su banco. Toda "petición de transferencia entre cuentas" debe enviarse al Banco que posee la cuenta receptora (donde esta la cuenta de la Tienda Virtual), para que éste certifique y asegure que la identidad del beneficiario (cuenta receptora = vendedor o atacante) declarado en la "petición", sea la misma que realiza la operación. Si esto no se cumple, el Banco rechaza la petición.

Si todo esta correcto, este Banco envía una "petición de transacción" en la cual adjunta la "petición de transferencia entre cuentas" en texto en claro, más el "certificado de identificación del beneficiario" al Banco poseedor de la cuenta emisora, por el medio interbancario.

El segundo Banco es quien:

1. Descifra la "autorización de Transferencia".
2. Verifica las firmas, "*timestamp*", integridad, monto autorizado y cuentas emisora y receptora de la "autorización de transferencia".
3. Coteja la información anterior con la de la "petición de transferencia entre cuentas" proporcionada.

Si TODO esta bien, los dos Bancos realizan la transferencia entre cuentas, le asignan un número de serie, la registran en sus respectivas bases de datos, y se genera un "acuse de transacción" para los bancos. El primer Banco envía "acuse de transferencia" al vendedor o emisor de la "petición de transferencia entre cuentas".

Si ALGO FALLA, se rechaza la "petición de transacción" y el segundo Banco le envía al primero el "rechazo de transacción", y éste envía su "rechazo de transferencia" al emisor de la petición. Además se pueden realizar otras acciones al analizar la falla:

- Si lo único que falla son las fechas:
    1. El reloj de la computadora del dueño de la cuenta emisora y consecuentemente, el "*timestamp*" de la autorización esta mal.
    2. El dueño de la cuenta receptora (tercero o vendedor) esta mandando una autorización antigua.
- Si no corresponden las cuentas receptoras de la autorización y la petición:
    3. El atacante quiere hacerse pasar por la cuenta beneficiada de la autorización.
- Si el Banco poseedor de la cuenta emisora descifra basura de la "autorización":
    4. Falla o no esta actualizado el "*crypto engine*" del dueño de la cuenta emisora.
    5. Existe un atacante haciendo pruebas, o quiere personificar al dueño de la cuenta emisora.
- Si lo que no corresponden son los montos en la "autorización" y en la "petición":
    6. El vendedor (cuenta receptora) trata de cargar más de lo pactado.
    7. El comprador (cuenta emisora) trata de pagar menos de lo pactado.
- Si fallan la firma y/o la integridad de la autorización:
    8. Existen fallas en la comunicación comprador-vendedor, pues se daño la autorización.
    9. El atacante modificó la autorización.

En las siguientes figuras se ilustran las arquitecturas SND del Banco, de la "Casa Musical" y de la "Tienda Virtual", respectivamente.

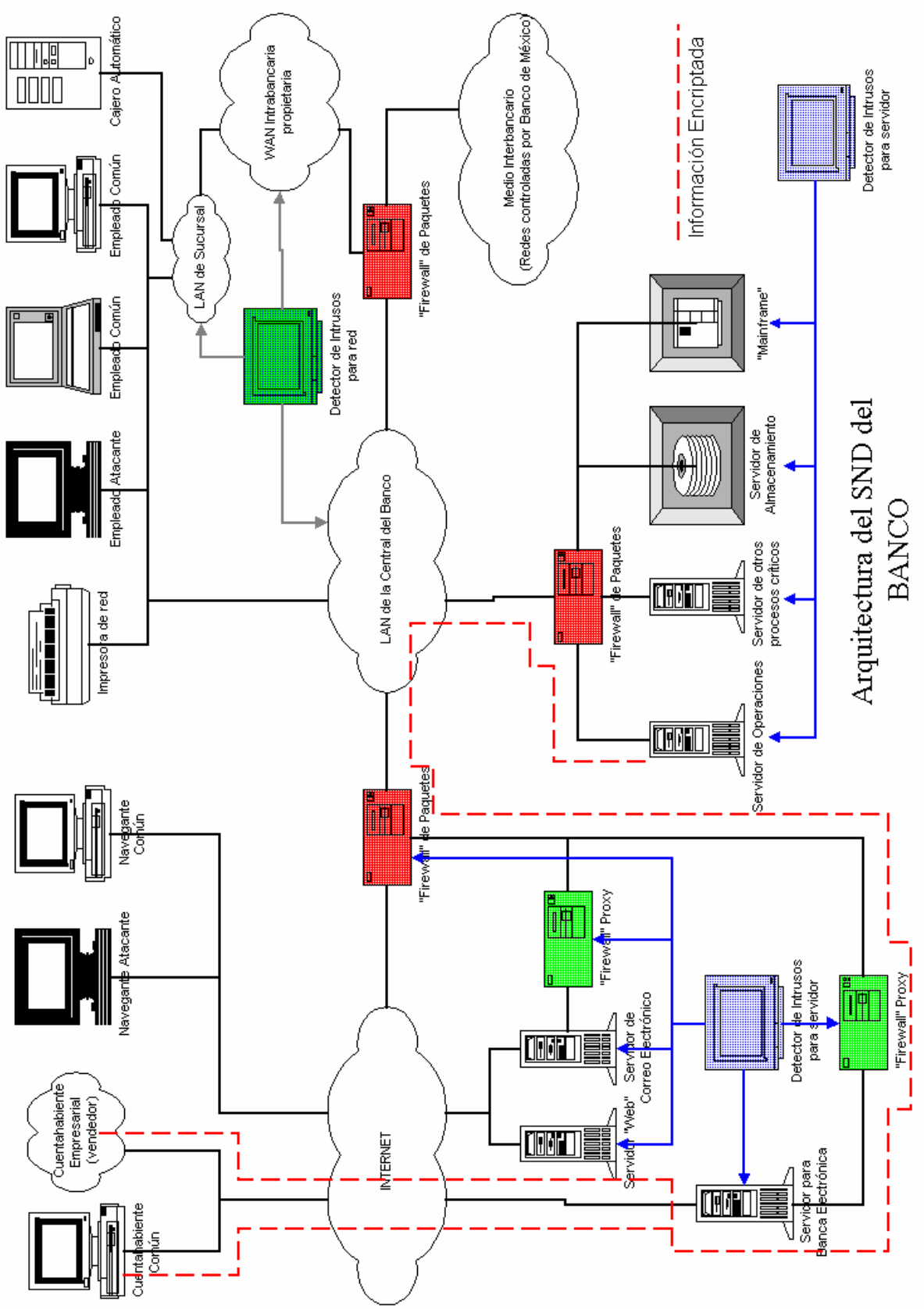


Figura 4-16. Arquitectura del SND para el Banco

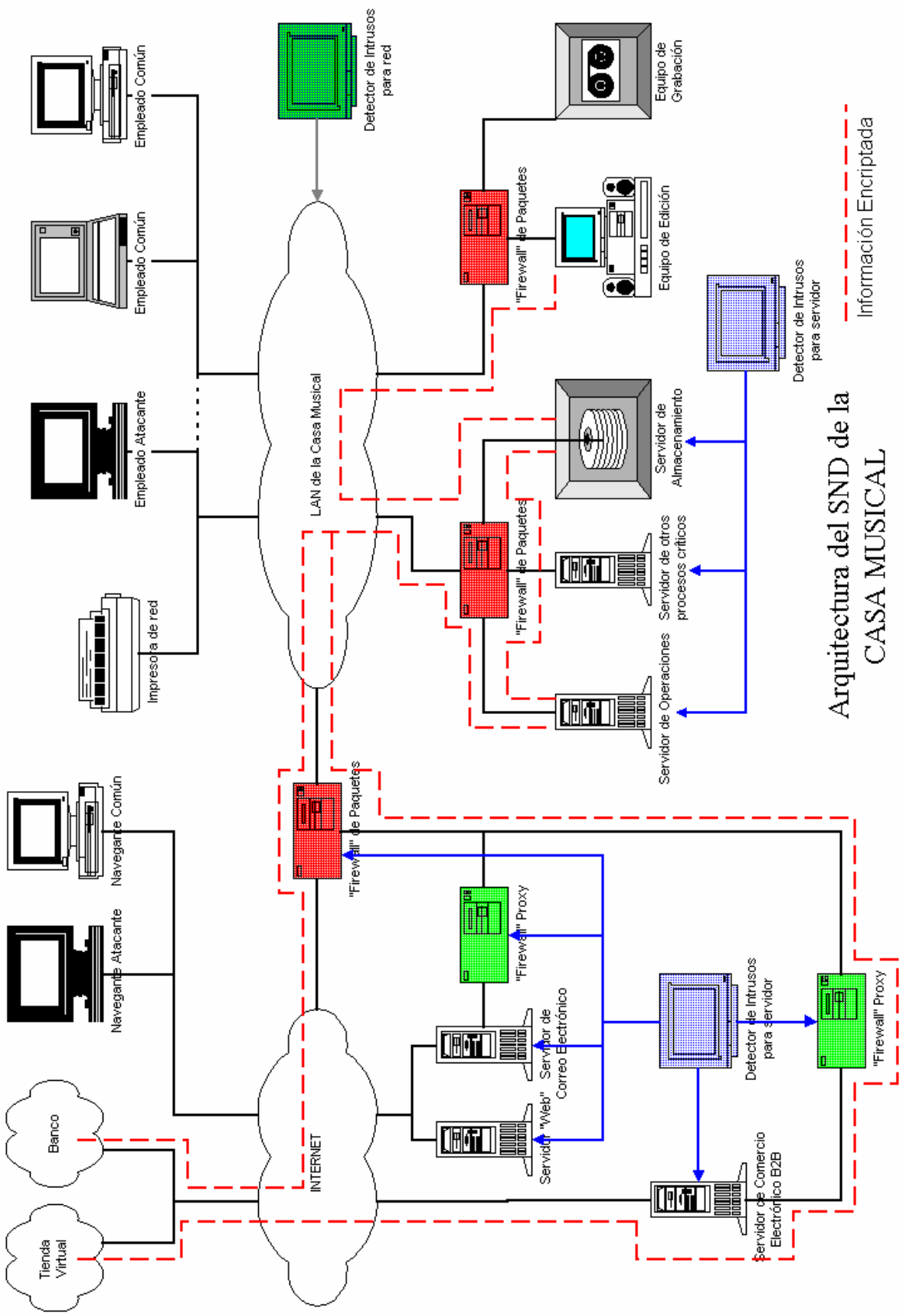


Figura 4-17. Arquitectura del SND de la Casa Musical

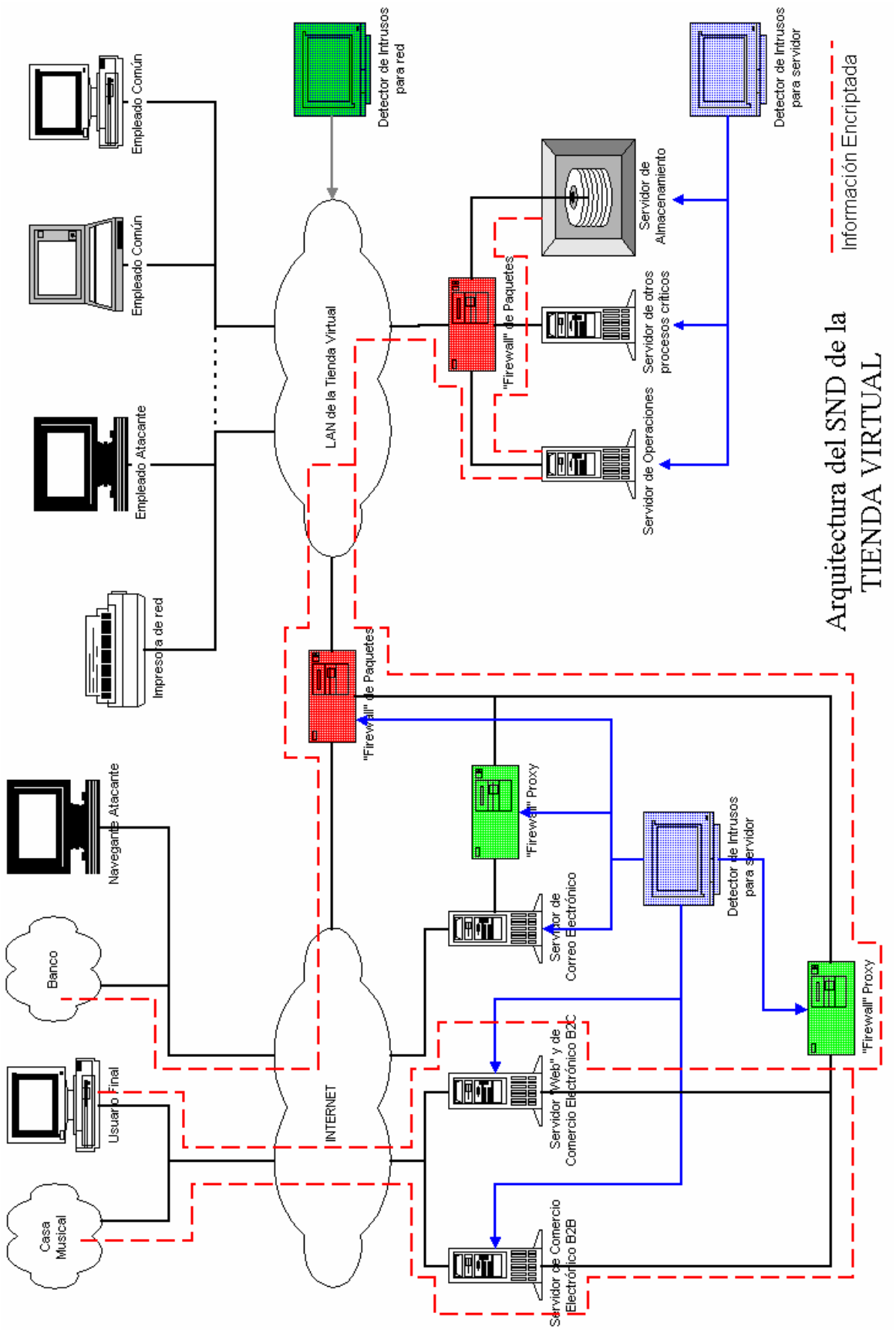


Figura 4-18. Arquitectura del SND de la Tienda Virtual

CAPÍTULO 5

EVALUACIÓN DEL DESEMPEÑO DEL MODELO DE COMERCIO ELECTRÓNICO Y RESULTADOS

Evaluación del Desempeño del Modelo de Comercio Electrónico y Resultados

En este capítulo se presenta los resultados más importantes correspondientes a las etapas y actividades realizadas durante el desarrollo del trabajo de tesis, así como el desempeño de los ejemplos desarrollados para el correcto funcionamiento del portal de comercio electrónico.

5.1 Java en el Comercio Electrónico

Al momento de hacer un análisis del portal de comercio electrónico a desarrollar y de observar las características que tenia que cumplir para ser considerado un Sitio *Web* de nivel III, se tomo la decisión de usar un lenguaje de programación, por medio del cual el usuario pudiera apreciar el uso de la criptografía en el comercio electrónico, utilizado tanto para las interfaces y programas, como para los algoritmos criptográficos necesarios.

Después de un estudio de los posibles lenguajes a usar, *Java* fue elegido por las siguientes características:

1. La existencia de librerías con funciones de encriptación para este lenguaje.
2. Capacidad para reproducir algunos tipos de formatos de audio, con características mejoradas; las cuales se utilizaron en el desarrollo del proyecto.
3. Operabilidad multiplataforma (para los sistemas operativos más populares), lo cual permite expandir el posible público usuario de la aplicación. Y también dicha característica lo ha convertido en uno de los pilares fundamentales para el desarrollo de aplicaciones para *Internet*.
4. Es un lenguaje orientado a objetos, lo cual facilita la implementación del modelo que se realiza.

Al momento de desarrollar el presente trabajo, aconteció que al descargar la versión del paquete de desarrollo del lenguaje *Java* del sitio *Web* de *Sun*: las librerías con las funciones criptográficas (agrupadas en el JCE) no estaban integradas a dicho paquete porque sólo pueden ser utilizadas dentro de los Estados Unidos (ya que la criptografía es considerada un arma por las leyes de aquel país). Estas disposiciones no sólo afectan a este lenguaje ni solo a esta compañía, sino que afectan a toda su industria enfocada a la seguridad informática.

Para el correcto funcionamiento de esta tesis, este escenario motivó a investigar posibles alternativas; encontrándose "JCE alternativos" desarrollados por grupos interesados en utilizar la protección de la criptografía con las capacidades de *Java*. Es un hecho que todas las "JCE alternativas" carecen de una documentación para utilizarlas; y en muchos casos carecen también de ejemplos, instrucciones y herramientas para instalarlas y probarlas. Sin embargo, un resultado concluyente de la investigación es la detección de un enorme potencial de desarrollo en este campo (en “*software*”, “*hardware*” e integración).

5.1.1 Implementación de Java

La siguiente etapa consistió en diseñar el sitio de comercio electrónico. Se inicio la creación de las páginas de la Tienda Virtual, y después se comenzó con el desarrollo de las bases de datos y la transferencia de la información entre ellas y las páginas del sitio, para dar al usuario mayor interactividad cuando lleva a cabo sus transacciones con el sitio; primero se utilizo una misma computadora como sitio de comercio electrónico y cliente, para lo cual se instaló el "*Personal Web Server*" para *Windows* 9x. Una vez en perfecto funcionamiento se traslado a un servidor de *Internet* como lo es el *IIS*, para poder así acceder de forma remota y probar el certificado de servidor y de este modo probar el canal seguro de comunicación entre el Servidor y una computadora personal remota, cabe hacer notar que tanto el servidor como la computadora, están dentro de la misma *Intranet*, ya que de otro modo se tendría que contar con un Nombre de Dominio válido para acceder al portal desde *Internet*, aunque para fines prácticos funciona de la misma manera.

Como se analizo en el capítulo anterior, fueron utilizadas las siguientes herramientas: *Access*, ASP, *Java*, *JavaScript* y HTML dinámico; para construir un sitio más interactivo con la información manejada. Casi todas las herramientas anteriores son tecnología propietaria de *Microsoft*, y algunas de las ventajas de este esquema fue la gratuidad de las herramientas, así como un desarrollo rápido por la fácil integración de las mismas (por proceder de la misma compañía); pero hizo que resaltaran las diferencias del lenguaje *Java* de Microsoft y el estándar de *Sun*.

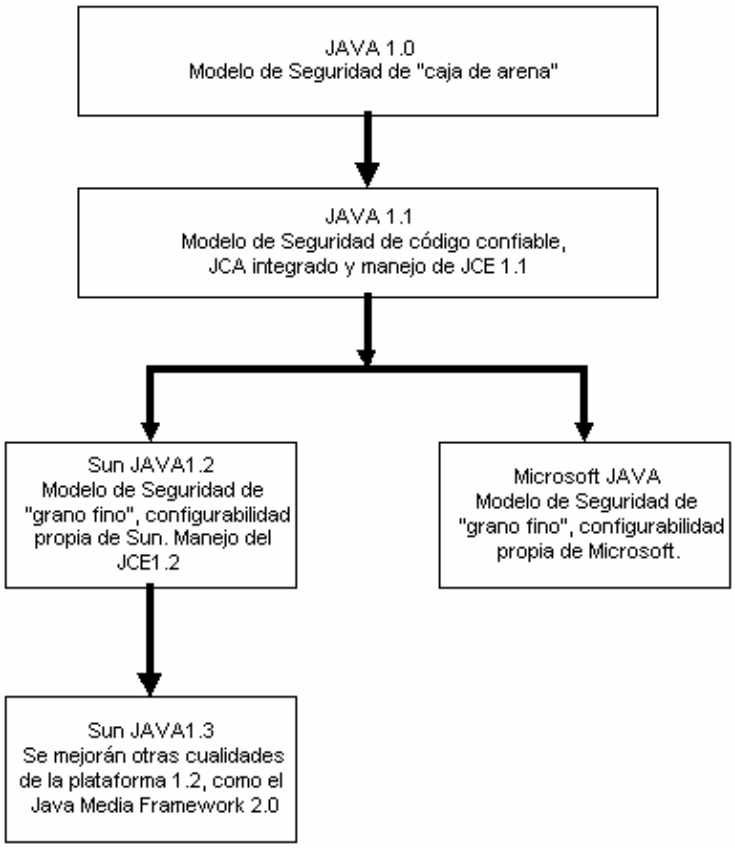


Figura 5-1. Evolución de Java y diferencias entre las implementaciones de Sun y Microsoft

El análisis de las dos implementaciones del lenguaje llevó a concluir que conceptualmente son iguales: las versiones más actuales manejan modelos de seguridad de “grano fino”; es decir, que pueden configurar y particularizar las medidas de protección a cada aplicación, o *applet* que el usuario utilice. Pero la forma en que se implementa dicho modelo varía drásticamente del esquema diseñado por *Sun* al utilizado por *Microsoft*.

El resultado más inmediato y útil fue encontrar que ambas compañías mantienen en común la compatibilidad con la plataforma *Java* 1.1, en la cual se incluyó el JCA y se manejó el JCE. Este hecho llevo a deducir que para diseñar aplicaciones, o *applets*, que requieran manejar las 2 plataformas, se debe de utilizar como eje de trabajo las clases definidas en la plataforma 1.1, o manejar clases de compatibilidad (lo cual se realizó en este proyecto con un “parche”), ya que al utilizar una plataforma mas actual hay una mayor incompatibilidad entre dichos modelos.

El futuro no es nada halagador ya que con el continuo desarrollo y especialización de cada vertiente hará más difícil una mayor generalización de las aplicaciones, por lo cual se ve la enorme necesidad de normalizar el lenguaje antes de que se presenten situaciones críticas. Una muestra de esto es el sistema operativo *Windows XP Service Pack 2 (SP2)* el cual ya no incluye el programa *Java Virtual Machine* por lo que puede que no se tenga instalada la aplicación Java necesaria para leer códigos y por lo tanto no funcionar adecuadamente nuestra aplicación; la explicación que dio *Microsoft* fue de que se veía desprotegido por medio de la *JVM* ante posibles amenazas (aunque muchos creen que es debido a la mercadotecnia entre estas empresas), sin embargo este se puede descargar gratuitamente desde http://java.com/es/ quien es el fabricante del programa, ya que un sin número de desarrolladores siguen utilizando a *Java* como su lenguaje predeterminado.

5.2 Algoritmos y Llaves Criptográficos para el Comercio Electrónico

En esta sección se analizaran los algoritmos y la longitud de las llaves criptográficas, con las necesidades del modelo de comercio electrónico que se ha desarrollado en la presente tesis.

Primero se muestran las características de fortaleza de los algoritmos y llaves, luego un análisis de los bienes informáticos a proteger; para después relacionarlos y extraer el conjunto de llaves y algoritmos a usar.

5.2.1 Fortaleza de Llaves y Algoritmos

La evaluación de la fortaleza de los algoritmos y llaves criptográficos han creado diversos criterios para evaluar unos y otros, pero es poco el desarrollo analítico aplicado, menor la divulgación de los resultados y no existe todavía una normalización para compararlos. De cualquier forma al criptoanalizar los algoritmos y llaves por medio de un ataque por fuerza bruta, se utilizan los resultados más demostrativos de los análisis y trabajos realizados por diferentes personas e instituciones.

En general, el ataque a cada familia de cifradores y llaves se estima en función de 2 parámetros: la capacidad de cómputo requerida para efectuar el ataque (la capacidad de cómputo generalmente se evalúa en mips-año: ejecutar un millón de instrucciones por segundo durante un año, lo cual significa ejecutar $3*10^{13}$ instrucciones) y la cantidad de dinero disponible por el atacante, aunque para el caso de algoritmos asimétricos, hay un tercer factor que se tiene que tomar en cuenta: el aumento de los algoritmos de factorización. En los últimos años, nuevos métodos de factorización de números primos ha hecho que claves asimétricas que se consideraban seguras en su tiempo ya no lo sean tanto en la actualidad.

Combinando estos factores se observa que si bien la velocidad de cálculo en ataques contra sistemas de clave simétrica aumenta un 70% al año, en el caso de sistemas de clave asimétrica el aumento es mayor: alrededor de 170%. Esto es importante, porque significa que las claves asimétricas se hacen vulnerables con el tiempo a mayor velocidad que las simétricas.

5.2.1.1 Comparación entre llaves asimétricas y simétricas

La comparación entre claves simétricas con asimétricas no es posible ya que las prestaciones y los modelos de ataque son tan diferentes que la comparación solo sirve para establecer un equilibrio de protección de las llaves de los algoritmos simétricos y asimétricos, en caso de que se diseñe y utilice un sistema híbrido (donde los asimétricos normalmente transportan las llaves simétricas y las simétricas protegen el almacenamiento de las asimétricas). La fortaleza de la llave de transporte debe ser por lo menos igual a la fortaleza de la llave transportada, y la llave de almacenamiento de las llaves de transporte debe tener una fortaleza superior a dichas llaves.

En la siguiente tabla [20], se observa una posible equivalencia de la longitud de las llaves para algoritmos simétricos y asimétricos.

Longitud (en bits) de llave simétrica	***Longitud (en bits) de llave asimétrica***
56	384
64	512
80	768
112	1792
128	2304

Tabla 5-1. Equivalencia de llaves simétricas y asimétricas

Debido a que mucha información del análisis de llaves esta sólo enfocada a simétricas o asimétricas; la tabla anterior permite encontrar la equivalencia de un dato específico, de un sistema a otro.

5.2.1.2 Llaves simétricas

Aunque no existe una forma de evaluar la robustez de cada algoritmo, es decir, sobre el como ha implementado los principios de confusión y difusión, ni cual es su eficiencia en el manejo de la longitud de la llave. Sin embargo, sí es posible evaluar cada cifrador en base a 2 parámetros:

1. La velocidad de encriptación (refleja indirectamente la confusión y difusión realizadas sobre el texto en claro; en [28] se puede ver con más detalle el criptoanálisis cuando se han implementado pocas o muchas etapas de confusión y difusión).
2. El desarrollo criptoanalítico enfocado en cada algoritmo particular; por la literatura [20] podemos ver que DES y sus variaciones (DES2X, TripleDES, etc.) han sido sometidos a un enorme trabajo criptoanalítico; lo cual obliga a no utilizar dicha familia de cifradores en aplicaciones de comercio electrónico actuales y futuras.

El ataque por fuerza bruta a estas llaves es la búsqueda en forma exhaustiva de la llave utilizada en el correspondiente espacio de llaves.

La ley de Moore establece que el poder de cómputo se duplica cada 18 meses, tiene por consecuencia directa que los costos de un equipo se reducen en un factor de 10, cada 5 años. Al aplicar esta ley a la Tabla 7.1 de [20], se relaciona la longitud de la llave con la inversión necesaria para romperla por medio de un ataque de fuerza bruta, haciendo el cálculo para los años 2000, 2005 y 2015.

Inversión 2000	Inversión 2005	Inversión 2015	40 bits	56 bits	64 bits	80 bits	112 bits	128 bits
$100 K	$10 K	$100	2 s	35 h	1 a	70,000 a	10^{14} a	10^{19} a
$1 M	$100 K	$1 K	0.2 s	3.5 h	37 d	7,000 a	10^{13} a	10^{18} a
$10 M	$1 M	$10 K	0.02 s	21 m	4 d	700 a	10^{12} a	10^{17} a
$100 M	$10 M	$100 K	2 ms	2 m	9 h	70 a	10^{11} a	10^{16} a
$1 G	$100 M	$1 M	0.2 ms	13 s	1 h	7 a	10^{10} a	10^{15} a
$10 G	$1 G	$10 M	0.02 ms	1 s	5.4 m	245 d	10^{9} a	10^{14} a
$100 G	$10 G	$100 M	2 μs	0.1 s	32 s	24 d	10^{8} a	10^{13} a
$1 T	$100 G	$1 G	0.2 μs	0.01 s	3 s	2.4 d	10^{7} a	10^{12} a
$10 T	$1 T	$10 G	0.02 μs	1 ms	0.3 s	6 h	10^{6} a	10^{11} a

Nomenclatura: s = segundos, m = minutos, h = horas, d = días, a = años

Tabla 5-2. Relación de la inversión necesaria para romper llaves simétricas

Suponiendo, según el ensayo "*Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commmercial Security*", que un atacante individual puede invertir en Dólares hasta $10,000; un atacante corporativo de $10,000 – $300,000,000 (según si la empresa es chica, mediana o grande); y un atacante gubernamental (como una agencia de inteligencia de Estados Unidos) de $300,000,000 en adelante.

5.2.1.3 Llaves asimétricas

Dado que los algoritmos de cifrado asimétrico se basan en operaciones con grandes números (factorización, multiplicación, exponenciación, logaritmos etc.); el principal

ataque por fuerza bruta para estos algoritmos es por la factorización de grandes números, y de acuerdo con las matemáticas actuales y el conocimiento criptográfico que se tiene se debe tener un especial cuidado con la longitud de llave a utilizar, por lo que al momento de elegir la clave asimétrica se tienen que seguir los siguientes principios:

1. Seguridad. Uno de los factores clave en la fortaleza de sistemas asimétricos es el tamaño de los pares de claves públicas / privadas.
2. Velocidad. Cuanto mayor es la clave, tanto más lentas serán las operaciones de clave pública.

Para el desarrollo de las llaves asimétricas que se utilizaron en esta tesis, el factor que se utilizo para la selección del tamaño de clave pública fue la *seguridad*, ya que la *velocidad* rara vez es un factor condicionante, debido a las siguientes razones:

1. El algoritmo de clave pública no se usa para el grueso del cifrado, sino sólo para cifrar la clave de sesión a cada destinatario.
2. Las computadoras son tan rápidas que la diferencia en rendimiento entre firmar o cifrar con una clave de 4096 bits y otra de 512 bits es prácticamente despreciable.

La siguiente tabla [31], lista las longitudes de clave pública recomendadas para protegerse contra ataques de acuerdo al presupuesto del atacante:

Año	Atacante Individual	Atacante Corporativo	Atacante Gubernamental
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Tabla 5-3. Longitud recomendada de claves asimétricas

5.2.2 Determinación de la Longitud de Llave Requerida

Como se ha mencionado en capítulos anteriores, los dos 2 bienes informáticos más importantes en nuestro modelo de comercio electrónico son la canción digitalizada y el dinero (número de tarjeta). Es por eso que es necesario analizar sus características para determinar las necesidades criptográficas a cubrir.

Para determinar la longitud de la llave requerida por cada bien, se deben considerar 3 parámetros:

1. El **tiempo de vida del bien**;
2. El **valor monetario del bien;**
3. El **tipo de atacante** a enfrentar, en especial por su capacidad económica.

Estos tres parámetros permiten determinar la longitud mínima de la llave (simétrica o asimétrica) que deberá ser usada para proteger dicho bien; ya que el tiempo necesario para romper la llave debe ser mayor que el tiempo de vida útil, y el costo de implementar el ataque debe ser mayor al valor del bien y el presupuesto del atacante.

5.2.2.1 Canción

Analizando la canción digitalizada, existen dos tiempos de vida importantes:

1. El **tiempo de novedad**, cuando la canción es nueva y es lanzada al mercado. Se presenta en todas las estaciones, sin importar género ni público. Este período puede ir de unas semanas hasta 2 años.
2. El **tiempo de popularización** por parte del público, cuando es asimilada como parte de la cultura musical de un grupo humano ("las mañanitas", "*happy birthday*", etcétera). Este período puede llegar a pasar los 100 años; y son pocas las canciones que alcanzan este nivel.

En cuanto al valor de las canciones, tienen dos evaluaciones:

1. El **precio de venta de la Casa Musical a la Tienda Virtual**. Es un precio alto, talvez hasta en US$10,000 según la popularidad del artista y/o aceptación de la canción.
2. El **precio de venta de la Tienda Virtual al Cliente Final**. Para este caso el precio oscila entre el costo de un CD (alrededor de US$15) y el precio de canción por CD (si en promedio un CD trae 15 canciones, cada canción cuesta US$1). Por el momento no se analizará el hecho de que dentro de un mismo álbum existen canciones exitosas las cuales tienen un precio diferente a las no exitosas.

Por último, el atacante al cual se enfrenta la canción puede ser de dos tipos:

1. **Atacante individual** o un grupo de atacantes individuales; los cuales cuentan con pocos recursos para invertirlos en equipo criptoanalítico.
2. **Atacante corporativo** (otra Tienda Virtual que desee la canción sin comprarla al alto precio de venta propuesto por la Casa Musical) con recursos moderados para realizar un ataque.

Por lo anterior, la llave a utilizar para proteger la canción debe tener una longitud que permita una protección de 100 años, un costo de ataque superior a los US$10,000; y soporte a un atacante tipo corporativo con recursos moderados.

Utilizando las tablas mostradas, se recomienda una llave simétrica de mínimo 80 bits de longitud (aunque para mayor seguridad una de 128 bits sería recomendable) o una llave asimétrica de mínimo 2048 bits de longitud; y no usar ningún cifrador de la familia DES. Dado que la canción va a ser reproducida de un almacén local y tiene un tamaño relativamente grande, se recomienda usar cifradores de bloque rápidos.

5.2.2.2 Número de tarjeta

Respecto al número de tarjeta (el cual conduce al dinero), se estima que éste puede tener un **tiempo de vida de hasta 60 años** (suponiendo que una persona a los 25 años abre su cuenta y la usa hasta su muerte), en el supuesto caso de que nunca cambie el número de la tarjeta asociada a la cuenta, y sea un cuentahabiente fiel al mismo banco durante ese lapso.

En cuanto al valor de dicho bien, esta relacionado directamente con tres cantidades:

1. El **monto de crédito disponible**,
2. La **cantidad depositada** en la cuenta, y
3. El **monto máximo permitido por día, por transacción o preestablecido por el cliente**, en las transacciones de comercio electrónico a cuenta habientes individuales.

Aunque existe un enorme rango de valores posibles para cada una de estas consideraciones, supondremos que el banco impone un tope máximo de US$300,000 (la compra de una casa) para operaciones de comercio electrónico, con sólo un número de tarjeta bancaria en un solo día.

Y los atacantes pueden ser de dos categorías:

1. **Atacante individual** o un grupo de atacantes individuales; con recursos moderados pero con un enorme interés de poder usar el dinero de otros.
2. **Atacante corporativo**, incluyéndose a las organizaciones criminales; lo cual nos lleva a considerar que pueden dedicar cuantiosos recursos para criptoanalizar el número de tarjeta.

Por lo cual se concluye que la llave a utilizar para proteger el número de tarjeta debe tener una longitud que permita una protección de 60 años, un ataque cuyo costo supere los US$300,000, y soporte un atacante corporativo con recursos cuantiosos.

Utilizando las tablas mostradas, se recomienda una llave simétrica de un mínimo de 112 bits de longitud (aunque sería mas conveniente una de 128 bits) o una llave asimétrica de un mínimo de 2304 bits de longitud. Dado que el número de tarjeta y la forma electrónica en sí son un conjunto de información relativamente pequeño, se pueden usar cifradores asimétricos.

5.2.3 Análisis de los Algoritmos a Usar

Dado que se van a utilizar aplicaciones y *applets Java* para implementar el modelo, se van a utilizar librerías JCE para realizar las operaciones criptográficas.

Se descartó el JCE de *Sun*, porque no puede ser descargado desde lugares fuera de los Estados Unidos, lo cual llevo a evaluar el JCE de Cryptix.

En cuanto al algoritmo de cifrador simétrico a usar, tenemos varios candidatos (todos presentes en la versión JCE y algunos en la 3.2.0):

- MARS
- RC6
- Rijndael
- Serpent
- Twofish
- Blowfish

Donde los primeros 5 concursaron para ser el algoritmo a usar como AES (*Advanced Encryption Standard*, siendo el ganador Rijndael).

Para el modelo, se decidió utilizar Blowfish como cifrador simétrico (dado que es un cifrador rápido, en la página de *Counterpane* se lista todas las personas y organismos que lo usan, es libre y no existe mucho trabajo criptoanalítico sobre él); y ElGamal como cifrador asimétrico (pues no existe tanto trabajo criptoanalítico en comparación con RSA). Sin embargo en el código del "*CryptoMusicMaker*", descrito en 5.5, también se considera usar Rijndael ya que fue el algoritmo seleccionado para el AES. En general, el manejo de algoritmos en el JCA y JCE de *Java* se resume a utilizar el nombre del algoritmo deseado en las funciones "*crypto engines*" creadoras de llaves y encriptadores / desencriptadores.

Por último, como función resumen se utiliza SHA-1 (disponible en las dos versiones de Cryptix) para las funciones de integridad de la información. Esto se debe a que SHA da un resumen de 160 bits, lo cual es más sensible a cambios en comparación a los 128 bits que ofrece MD5.

5.3 Ejemplos Desarrollados

Los ejemplos que se muestran a continuación son páginas "*Web*" desarrolladas para evaluar las implementaciones de los cifradores antes de su uso en el portal comercial desarrollado. La idea al desarrollar estas páginas, fue la de analizar la forma en la que operan los "*applets*" ya creados, y analizar la forma en la que están definidos sus constructores y métodos del lenguaje *Java*, para de esta forma, efectuar las modificaciones necesarias en ellos de acuerdo a las características fundamentales de nuestro problema y, además, puede servir como material didáctico en posteriores cursos o como una base para tesis que intenten desarrollar un tema similar.

5.3.1 Cifrador de Cesar

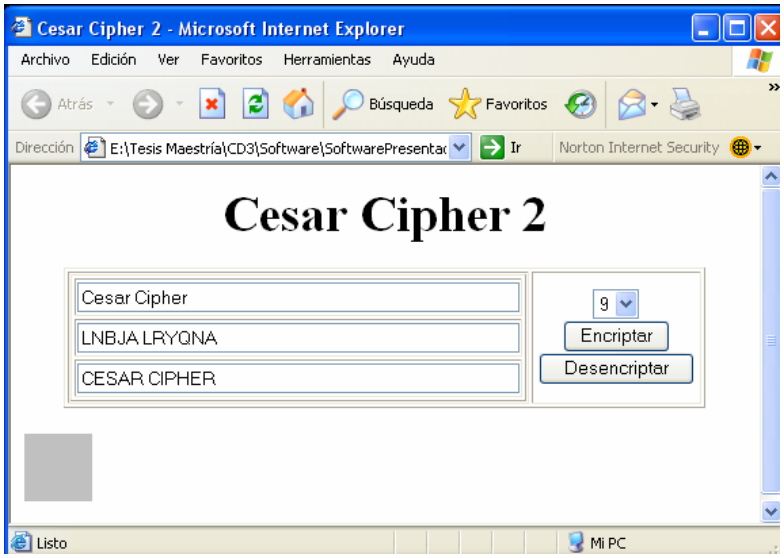


Figura 5-2: Cifrador de Cesar

El primer algoritmo desarrollado fue el Cifrador de Cesar. Como se puede apreciar en la figura 5-2, se tienen tres cajas de texto, de las cuales, en la primera el usuario introduce cualquier texto, al pulsar el botón de “Encriptar”, aparece en la segunda caja el texto cifrado según el número especificado en la caja de selección. Hay que recordar que el cifrador de Cesar es uno de los métodos más sencillos (y antiguo) de encriptación. Su forma de operar es básicamente sustituyendo cada letra por la letra correspondiente “n” caracteres más adelante (o hacia atras) en el alfabeto. Para el ejemplo, al cifrar la palabra “Cesar Cipher” la primera letra C se permuta por L debido a que esta última esta 9 posiciones más adelante en el alfabeto.

Posteriormente, al pulsar el botón de “Desencriptar” aparece en la tercera caja de texto el mismo texto pero después de haber aplicado la operación inversa.

5.3.2 Páginas con Funciones Resumen

Para el segundo ejemplo, mostrado en la siguiente figura, se hizo algo similar. En este caso, el usuario escoge una cadena de texto de las que se muestran en la tabla de ejemplos iniciales. Introduce esta cadena en la primera caja de texto y el “*applet*” se encarga de buscar la función resumen de dicha cadena. En este caso, se puede escoger entre los algoritmos SHA-1 y MD5. Esta aplicación se utilizo como función resumen de los datos que ingresa el cliente al momento de realizar una compra en la pagina *Web*, y con dicha función resumen se crea el *password* para identificarlo la próxima vez que realice una compra.

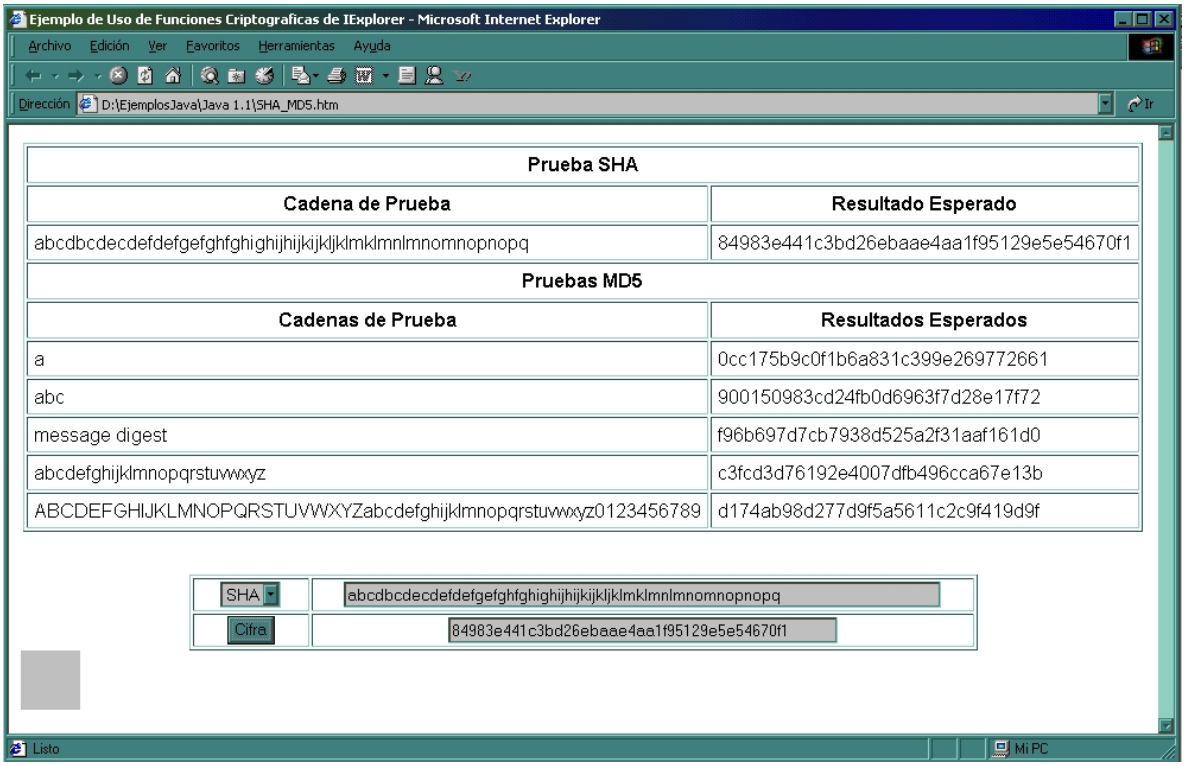


Figura 5-3: Funciones Resumen

Cabe hacer notar que la arquitectura básica de seguridad de *Java* posee dos elementos esenciales la Arquitectura Esencial de Seguridad de *Java* y la Arquitectura Criptográfica de *Java* (JCA). Dentro de la segunda, están definidas las clases que precisamente permiten obtener la función resumen de cadenas de texto de cualquier longitud. Estas clases fueron utilizadas para la implementación de esta página.

5.4 Sitio "Web" desarrollado

El siguiente paso fue el desarrollo de nuestro portal *Web* y entender todo lo que implica el entrar al comercio electrónico, se diseño un pequeño sitio con tecnología *Microsoft*: *Windows* XP, IIS, *Access* (para las bases de datos), DHTML y ASP para traer y presentar los datos en forma rápida e interactiva; así como también se utilizo *Java* y *Java* Script para ciertas funciones e interfaces. En el capítulo 4 se describe detalladamente el uso de estas tecnologías.

El simulador programado tiene una funcionalidad que permite ver y analizar el comportamiento del sitio *Web* y las actividades que un comprador cibernético podría hacer en él.

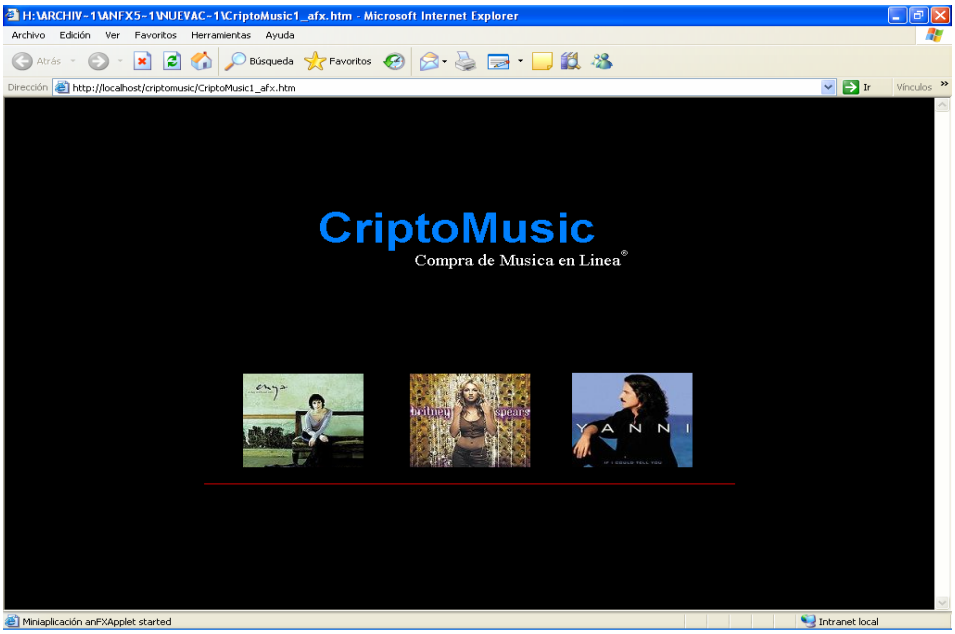


Figura 5-4. Presentación de la Tienda Virtual.

En la siguiente imagen se muestra la página donde el usuario puede visualizar y seleccionar las canciones que desee comprar, así como una breve descripción de las canciones disponibles en el catálogo.

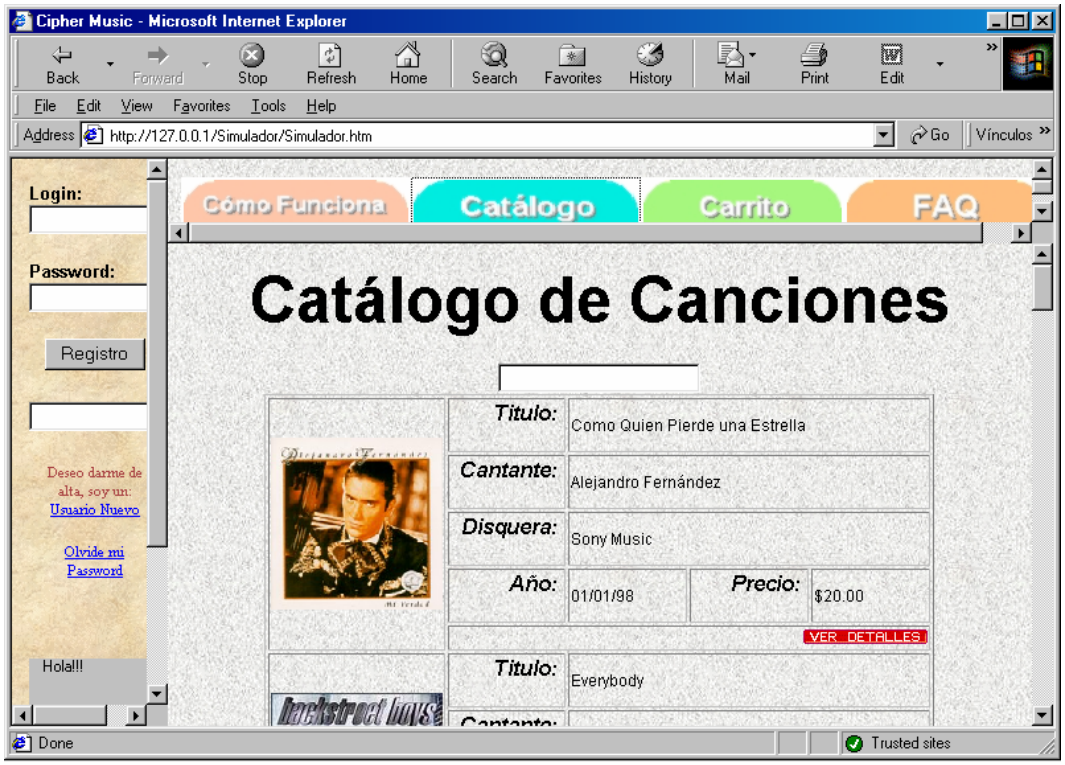


Figura 5-5. Página que muestra el catálogo de la Tienda Virtual.

También se pueden observar las características que tiene la Tienda Virtual, donde el cibernauta entra, se registra (mediante el uso de funciones resumen), escoge las canciones que le interesan y puede ver con el botón “Ver Detalles” toda la información del bien que desea adquirir antes de comprarlo, así como los diversos menús que explican en detalle el funcionamiento del portal electrónico.

De igual forma, se construyó la página donde el usuario puede ver su “carrito de compras”; en la cual se enlistan todas las canciones seleccionadas por el cliente cibernético y tiene la opción de “quitarlas” antes de enviar su pedido a la Tienda Virtual.

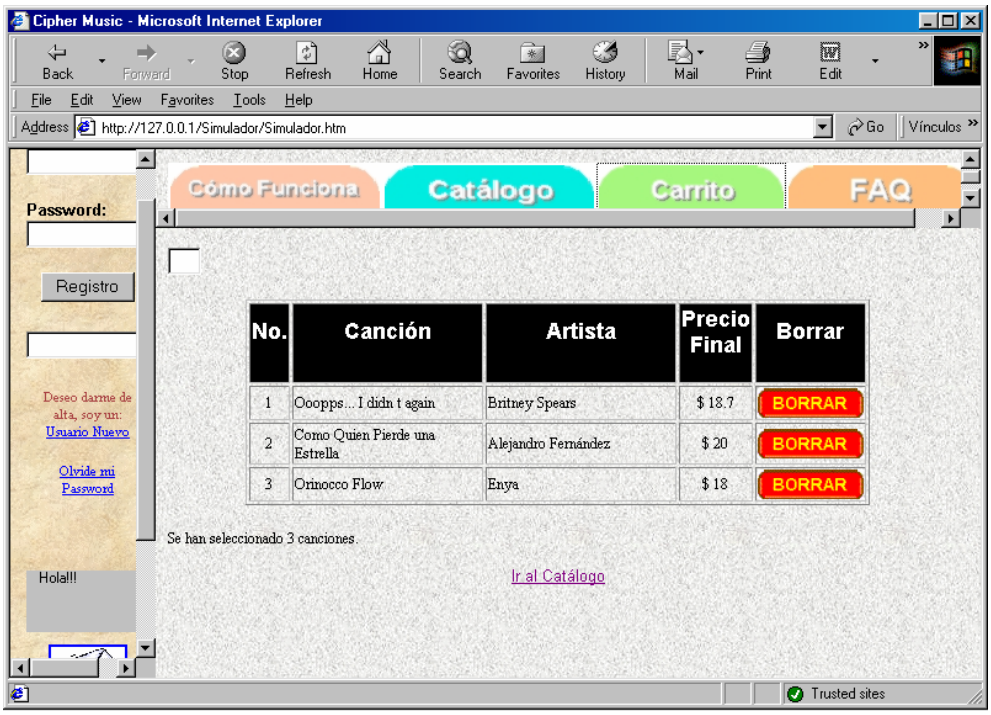


Figura 5-6. Página que muestra el Carrito de compra de la Tienda Virtual.

Posteriormente para poder implementar todas las características que requiere un Sitio *Web* de nivel III, para esto se tuvo que efectuar la instalación de un certificado digital ya que para que un sitio de comercio electrónico tenga éxito la seguridad debe ser confiable; la siguiente imagen muestra la pagina *Web* que se configuró para que contenga una conexión segura a través de *Internet*, por medio del protocolo SSL, gracias a este protocolo la información de la tarjeta de crédito que se envía entre el cliente y el servidor permanece segura.

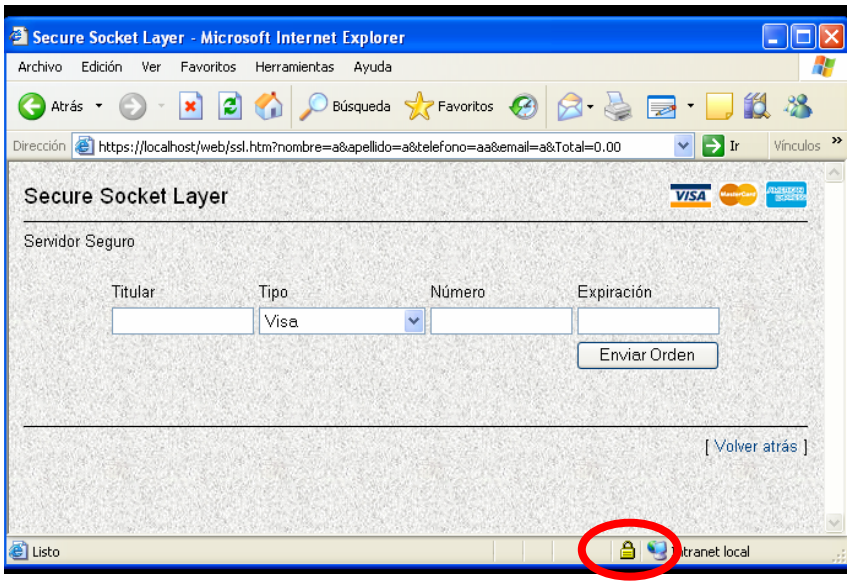


Figura 5-7. Página que muestra el certificado digital mediante SSL.

5.5 "CryptoMusicMaker"

El manejar herramientas criptográficas dentro de páginas "*Web*", a diferencia de aplicaciones para sistemas locales, llevó a evaluar muy diversas alternativas (ya que no existe documentación ni ejemplos al respecto). Observando los ejemplos presentados en la sección 5.3, el 'Cifrador de Cesar' representa la concepción de crear todo el esquema criptográfico a partir de cero. Se tuvo que programar todo el funcionamiento en *Java* y *JavaScript*, y tal vez sea una buena solución pero tiene el inconveniente de ser lenta (sobre todo al implementar un algoritmo robusto como Blowfish, AES, RSA, etc.). Por tal motivo se empezó a recurrir a las herramientas y arquitectura criptográfica ya desarrolladas del lenguaje *Java*, donde el ejemplo de la página con Funciones Resumen muestra el manejo de las herramientas de "*hash*" para mensajes que uno escriba en un campo de la página.

Finalmente, después de mucho análisis y pruebas, se logró ejecutar dentro de un "*applet*" las librerías de encriptación del JCE de Cryptix. Esto era necesario para crear el ejemplo de la concepción básica del "*CryptoMusicMaker*", donde una página "*Web*" es usada para cifrar archivos con extensión "***.txt***" utilizando ya sea, un algoritmo de cifrado DES, Blowfish, o Rijndael. La página HTML se muestra en la siguiente figura:

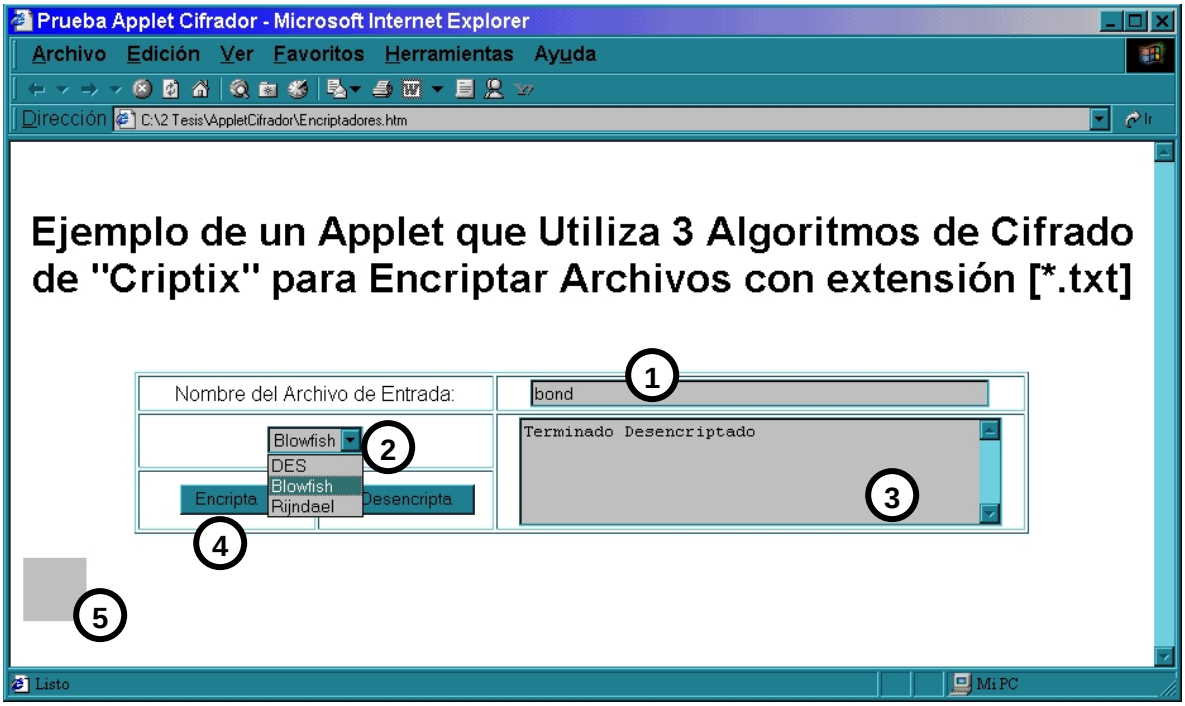


*Figura 5-8: Encriptador de Archivos *.TXT*

De acuerdo con la figura 5-8, se puede ver que el funcionamiento de la página es como se explica a continuación:

A) Primero, en la caja de texto de entrada (1), se escribe el nombre del archivo que se desea cifrar (sin extensión), sabiendo que este deberá tener extensión ***.txt*** ; en realidad, no es necesario que los archivos a cifrar sean de tipo texto, puede usarse cualquier archivo y únicamente será necesario cambiar su nombre, o extensión.

B) A continuación se escoge el algoritmo mediante el cual se desea cifrar usando la caja de selección (2).

C) Los botones de “Encripta” y “Desencripta” (4) son disparadores de los métodos de Encriptado y Desencriptado del “*applet*” (5).

D) Debe hacerse notar que, al finalizar las tareas, el usuario observaría en la ventana de respuestas (3) una serie de mensajes cortos en los que se le informaría de la conclusión de cada una de las tareas. Dado que estos mensajes son en algunos casos limitados, también puede programarse la consola de *Java* para enviar mensajes de control e informes del progreso del encriptado-desencriptado tal y como se muestra en la figura 5-9.

Aún cuando en este documento no se muestra el código de este ejemplo, las librerías de “*Cryptix*” permiten configurar también el modo de operación de los algoritmos de encriptación (dado que estamos trabajando con cifradores de bloque), la longitud en bits de la llave y el relleno, o complemento (“*padding*”).

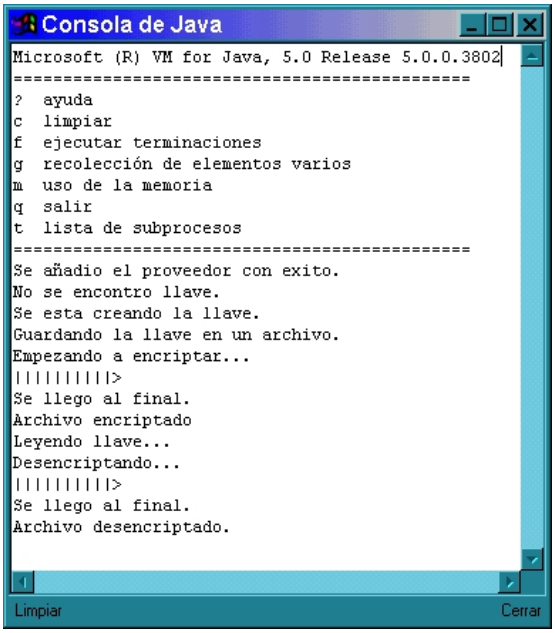


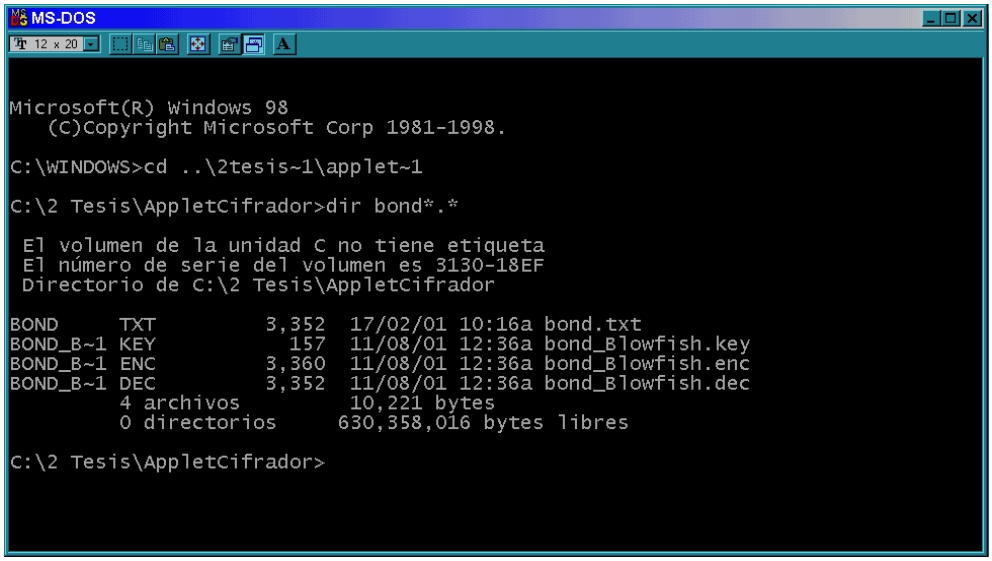
Figura 5-9. Consola de Java con los resultados del encriptado-desencriptado.

En esta imagen puede verse que el primer proceso que debe efectuar el programa es cargar el proveedor de encriptado (“*Criptix*”) para que, a través de él, pueda hacer uso de los algoritmos de cifrado (DES, Blowfish, Rijndael), y el resto de las herramientas criptográficas que maneja este paquete. Ya con el proveedor listo, se verifica la existencia de la llave adecuada. En el caso de que no exista la llave, esta se genera y almacena en el disco duro.

El siguiente paso es precisamente el encriptado (o desencriptado). El resultado final, tal y como se muestra en la figura 5-10, será tanto el archivo original (con extensión ***.txt***), más tres archivos adicionales. El primero de ellos tiene extensión ***.key*** y se refiere a la llave

usada por el algoritmo; el archivo encriptado tendrá extensión ***.enc***. Debe hacerse notar que, dado que se efectúa un relleno (*padding*) el tamaño de este archivo es, por lo regular, algunos bytes mayor que el archivo original. El último archivo se identifica por la extensión ***.dec*** y es el archivo desencriptado.

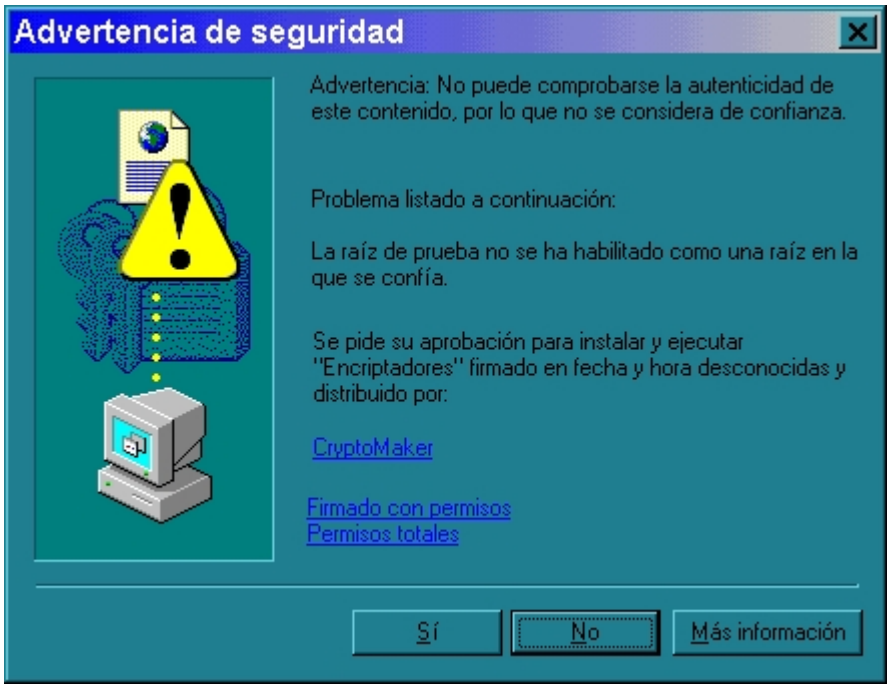
Algo que debe quedar muy claro es que, para efectuar esta simulación, el "*applet*" debe tener la capacidad de leer y escribir archivos en el disco duro de la máquina local. Para dotar al "*applet*" de los permisos necesarios para efectuar estas operaciones, primero es necesario generar un certificado digital y "firmar" electrónicamente el código que formará parte del programa. Desafortunadamente no existe mucha literatura respecto a la forma en la que deben efectuarse estas operaciones; sin embargo, se recomienda leer la página de *Internet* marcada con la referencia [35] de la bibliografía, así como la ayuda en línea del SDK de *Java* creado por *Microsoft* [52].



5-10. Terminal de MS-DOS. Pueden verse los archivos creados con sus extensiones y tamaño en bytes

Además, el costo de los certificados digitales no es bajo; razón por la cual se decidió crear nuestros propios certificados con los cuales se trabajó durante la etapa de desarrollo (el JDK de *Microsoft* así como el de *Sun* tienen herramientas propias para crear este tipo de certificados "no legales").

Cuando un usuario cargue la página HTML por primera vez, aparecerá la advertencia mostrada en la figura 5-11.



5-11. Certificado Digital

Este mensaje indica que el "*applet*" que se ejecutará en la página esta diseñado para efectuar operaciones que "pueden ser consideradas peligrosas"; tal como escribir en el disco duro. Por esta razón, si cualquier usuario ve una advertencia similar a ésta al navegar por *Internet*, deberá estar seguro de que, quien escribió un "*applet*" con estas características, es un programador de confianza, pues de otra forma, su información puede verse comprometida.

Para entender mejor cuales acciones son consideradas por *Java* como "peligrosas" es recomendable leer el anexo _ "Modelo de Seguridad de *Java*".

Cuando los sitios en *Internet* cuentan con este tipo de certificador pero dichos certificados han sido emitidos por entidades comerciales de confianza (tales como *Verisign* o *e-Trust*), el certificado se instala en el navegador del cliente de forma transparente. Esto se debe a que, al ser entidades de confianza han establecido de antemano acuerdos y licencias con los productores de "*software*" (*Microsoft*, *Netscape*, etc.), de forma tal que los certificados que estas empresas emiten están listos para aplicaciones de comercio electrónico.

5.6 "Crypto Player"

La última aplicación desarrollada fue el "*Crypto Player*": una aplicación para ejecución en sistemas locales, donde utilizando las capacidades multimedia de la arquitectura *Java2*, se pudiera ejemplificar el uso de la criptografía para proteger bienes informáticos, vendidos a través del comercio electrónico y sujetos a ser atacados por la piratería.

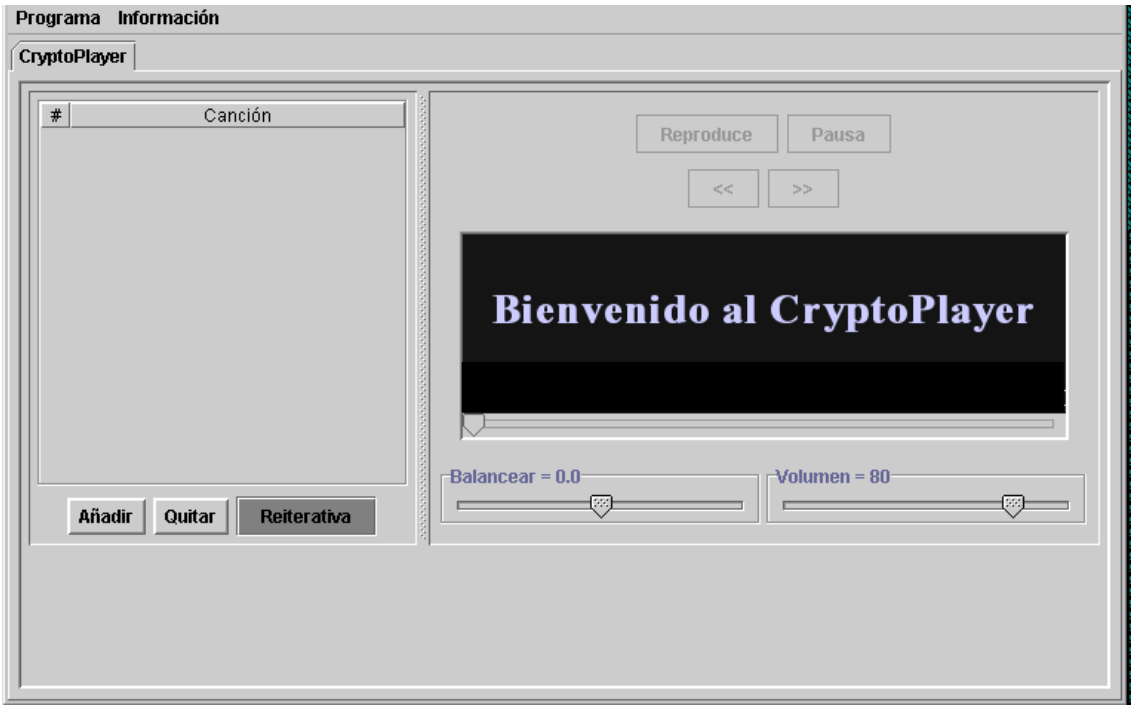


Figura 5-12. Pantalla de bienvenida del "Crypto Player"

Esta aplicación se basa en el ejemplo "*JavaSound*" provisto en el JDK2 edición Estándar de *Sun*. Únicamente se exploró la parte criptográfica para mostrar como un bien informático protegido criptográficamente solo puede ser utilizado por las herramientas permitidas (véase figura 5-13).

Aunque como se mostró en el modelo teórico propuesto en el capítulo 2, para que dicha aplicación sea comercialmente aceptable, debe manejar "*passwords*", así como particularizarse al sistema local en el cual se ejecuta. Pero el desarrollo de los procesos de personalización, instalación, particularización de los recursos detectados del usuario, ocultamiento de información vital en diversos lugares del sistema y protección propia del "*software*" contra la piratería; son elementos que salen fuera del alcance del presente trabajo de tesis.

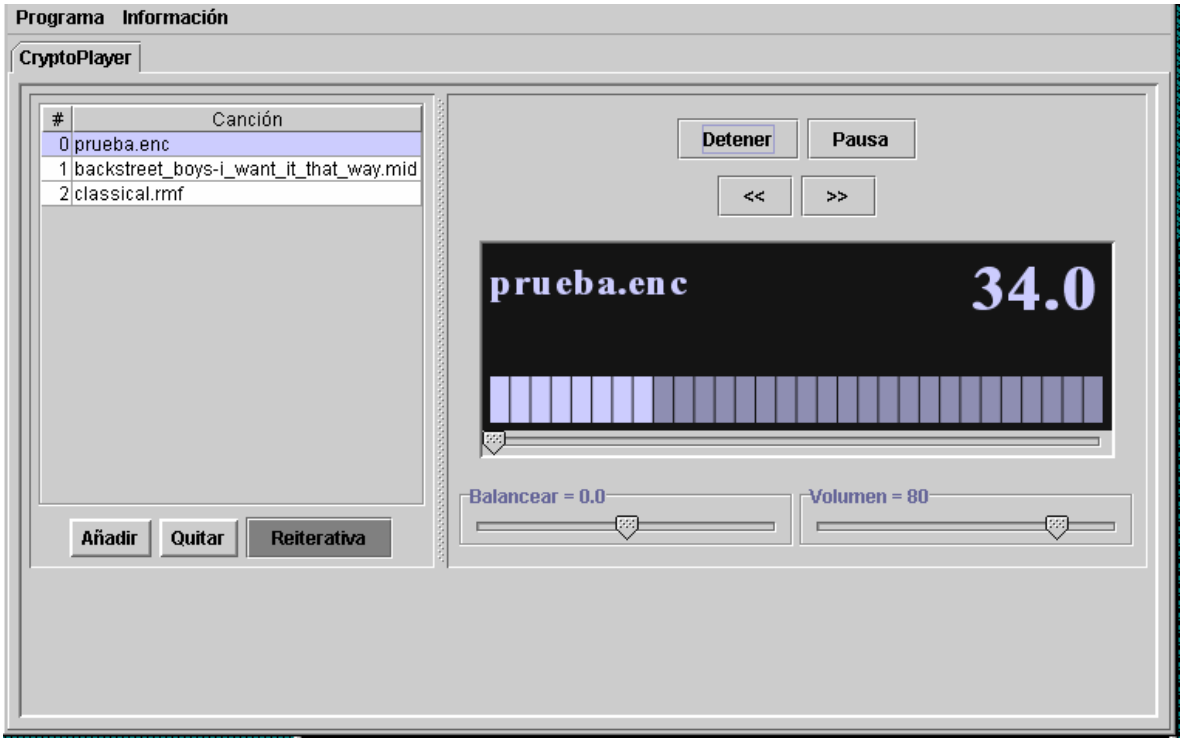


Figura 5-13. Ejemplo de reproducción del "Crypto Player"

CONCLUSIONES

Conclusiones

El comercio electrónico nace como una alternativa de reducción de costos, y como herramienta fundamental en el desarrollo empresarial; formando parte importante en el mundo de *Internet*; hace sencilla la labor de negocios, reduce costos y precios, y garantiza una disponibilidad las 24 horas del día. Sin embargo, elimina el contacto directo y por ende parte del conocimiento de la empresa y el cliente; así como también crea desconfianza en cuanto a la seguridad del sistema. En términos generales debe ser entendido como el acto de compra venta de toda clase de bienes y servicios que utiliza medios de telecomunicaciones para su realización incluyendo el pago por esa misma vía.

Al igual que en el comercio tradicional cada forma de comercio que se ha inventado está sujeta a un riesgo y el comercio electrónico al realizar transacciones por *Internet* no esta exento, el comprador teme por la posibilidad de que sus datos personales (nombre, dirección, número de tarjeta de crédito, etc.) sean interceptados por "alguien", y suplante así su identidad; de igual forma el vendedor necesita asegurarse de que los datos enviados sean de quien dice serlos. Sin embargo, estos sistemas son exitosos debido a que sus beneficios y conveniencias sobrepasan las pérdidas.

Los esquemas de comercio electrónico presentan alguna forma de falsificación, tergiversación, negación de servicio, o cualquier otro tipo de trampa. De hecho, el avance en la computación ha permitido que el riesgo aumente, al habilitar ataques antes imposibles. La información, y en especial las malas noticias, se mueven demasiado rápido: una debilidad en seguridad que sea descrita en *Internet* puede ser aprovechada por miles antes de que se encuentre una solución. Los sistemas de seguridad no son perfectos, pero usualmente son suficientes para nuestras necesidades, e incluso hoy deben anticiparse a ataques futuros.

El principal problema para un mejor desempeño del comercio electrónico es la confiabilidad que este maneja para que tenga éxito, ya que es un hecho que el comercio electrónico no ha experimentado el crecimiento ni la aceptación que el entusiasmo inicial pronosticaba para el futuro inmediato, aunque también es cierto que el número de personas que compran en línea ha crecido bastante en los últimos 2 años. Existen muchos aspectos abiertos en torno al comercio electrónico; entre ellos se pueden destacar, la validez de la firma electrónica, no repudio, la legalidad de un contrato electrónico, las violaciones de marcas y derechos de autor, pérdida de derechos sobre las marcas, pérdida de derechos sobre secretos comerciales y responsabilidades.

A pesar de los intentos por dar mayor seguridad, tanto con los avances técnicos como con los legislativos, los usuarios no acaban de estar convencidos, es decir, el proceso de aceptación está llevando más tiempo del previsto. Al desarrollar este trabajo de tesis, se encontró con la posibilidad de ampliar un mapa de un nuevo territorio en el cual se conjuntan la seguridad, la criptografía y el comercio electrónico. Su creación requirió el estudio y análisis de diversos temas: la información en formato *bit*, el comercio electrónico, la seguridad de la información, la criptografía, el diseño de páginas y servicios de *Internet*, y el uso de metodologías y lenguajes de programación para diseñar sistemas informáticos seguros.

Durante la investigación de las herramientas necesarias para la creación de un sitio de comercio electrónico seguro se observó la necesidad de indagar diferentes técnicas de seguridad, las cuales principalmente son complementarias unas de otras. Se analizaron desde mecanismos criptográficos hasta métodos de seguridad física, para poder cubrir los requisitos de seguridad necesarios se requirió cubrir varios aspectos, ya que si bien es importante disponer de un canal seguro para proteger las transacciones que se efectúen sobre él, este no serviría de nada si cualquier persona pudiera tener acceso directo al servidor encargado de almacenar la información.

Entorno a los temores fundamentales que se presentan al trabajar en el comercio electrónico, se hizo uso de diferentes herramientas criptográficas fundamentalmente. Se realizo la creación de un certificado digital mediante el protocolo SSL el cual permitió implementar canales de comunicación seguros a través de *Internet*, especialmente para la realización de transacciones comerciales, para ello se hizo necesario disponer de un servidor *Web* que permitiera instalar el certificado, toda la información que es transmitida por este canal es confidencial, cifrada y viaja de forma segura; esto brinda confianza tanto a proveedores como a compradores que hacen del comercio electrónico su forma habitual de negocios.

Por otro lado se realizaron diferentes tipos de pruebas de *software* para apreciar cuales ofrecían mejores herramientas criptográficas, para poder diseñar las aplicaciones necesarias que permitieran cifrar la información a vender y que esta a su vez solo pudiera ser reproducida por el comprador imposibilitando su “clonación”, mediante el uso de algoritmos simétricos, algoritmos asimétricos y funciones resumen (*hash*). De lo anterior se llegó a la conclusión que el lenguaje Java ofrecía una serie de librerías que permitían desarrollar mas claramente estos principios sin embargo se tuvieron varios inconvenientes, uno de ellos fue que al tratar de bajar las librerías criptográficas (JCE) de la pagina oficial de *Sun* estas no se podían descargar, ya que la legislación de los Estados Unidos ha limitado a las empresas de dicho país a exportar sus productos criptográficos; y en general, la expresión del conocimiento e información en ésta área. Razón por la cual existe un gran vacío que reclama metodologías de aplicación así como productos en *software* y *hardware*, por lo que es muy importante empezar a desarrollar metodologías de aplicación criptográfica, las cuales conducirán a desarrollar *software* y / o *hardware* criptográfico seguro, y en general, aplicaciones e infraestructuras informáticas seguras.

Por lo anterior se recurrió a buscar JCEs alternativos, encontrando el de Cryptix como una buena opción ya que maneja librerías muy parecidas al JCE de Sun. Cryptix presenta una buena implementación pero es muy escasa su documentación para instalarlo y usarlo, en especial porque la poca que existe se enfoca más a los sistemas UNIX y hablan poco de los posibles problemas o “peculiaridades” de Windows.

También mediante Java se crearon *applets*, utilizados para la conceptualización del comercio electrónico los cuales requerían ciertos “privilegios” para poder ser ejecutados por lo que se hizo necesario crear certificados de autentificación mediante herramientas propias de Java, este certificado se realizo para fines de estudio ya que al no ser emitido por una Autoridad Certificadora el usuario al abrir una pagina *Web* con *applets* ve una advertencia de seguridad, la cual se puede inhibir instalando el certificado en cada computadora en que se utilice; para un red interna esto es posible, pero al ser este para un sitio de *Internet*, es una tarea imposible de realizar.

Por último se logró desarrollar un sitio de comercio electrónico en el que se pudiera probar el funcionamiento de compra-venta de canciones encriptadas por *Internet*, en él se utilizó todos los mecanismos propuestos como un sistema informático seguro; De igual forma se programó una aplicación que desencripta la canción comprada para reproducirla. Es conveniente hacer notar que el hecho de que se haya escogido las canciones como producto para vender, el desarrollo de esta tesis es apto para la venta de cualquier producto, dado que en cualquier otra aplicación se requerirá la utilización de los mismos mecanismos y grados de protección.

Las pruebas son una actividad esencial para la puesta en funcionamiento del comercio electrónico. Garantizan la confiabilidad del *software* y la seguridad del sistema. Cada elemento implicado en un sistema de comercio electrónico es sometido a un riguroso proceso de prueba. Este proceso asegura la existencia de un sitio confiable de comercio electrónico que crea a su vez confianza en el cliente.

Este documento intenta contribuir a despertar el interés en la investigación de temas relacionados con la seguridad informática y el comercio electrónico; tanto por el potencial económico que representa como por la necesidad de contar con mecanismos de seguridad específicos para cada empresa y/o gobierno, evitando con esto la realización de fraudes electrónicos. Es además una contribución concreta al desarrollo de esta área de la ingeniería.

ANEXOS

Anexo A: Protección en las Comunicaciones

En teoría, cuando personas desean comunicarse de forma segura, la encriptación puede ocurrir en cualquier capa del modelo de comunicaciones OSI (“Open Systems Interconnection”, Interconexión de Sistemas Abiertos). En la práctica, éste proceso se realiza en las capas más bajas o en las más altas. Si ocurre en las capas inferiores, se le llama **encriptación de enlace por enlace** (“link-by-link encryption”), donde todo lo que viaje por ese enlace de datos particular es encriptado. Si la protección de la información se efectúa en las capas superiores, se le llama **encriptación de extremo a extremo** (“end-to-end encryption”); los datos son cifrados selectivamente y permanecen en dicho estado hasta que son descifrados por el receptor final a quien fueron enviados. Cada una de estas dos alternativas tiene ventajas y desventajas.

A.1 Encriptación de Enlace por Enlace

El lugar más fácil para añadir la encriptación es en la capa física. Las interfaces a la capa física por lo general son normalizadas y es fácil conectar artefactos de encriptación (“hardware”) en este punto. Estos dispositivos encriptan toda información que pase a través de ellos, incluyendo datos de usuarios, información de enrutamiento e información de protocolos. Y pueden ser utilizados en cualquier tipo de enlace digital de comunicación. Por otro lado, los nodos de almacenamiento o conmutación inteligente entre el transmisor y el receptor necesitan descifrar el flujo de datos antes de procesarlo [20].

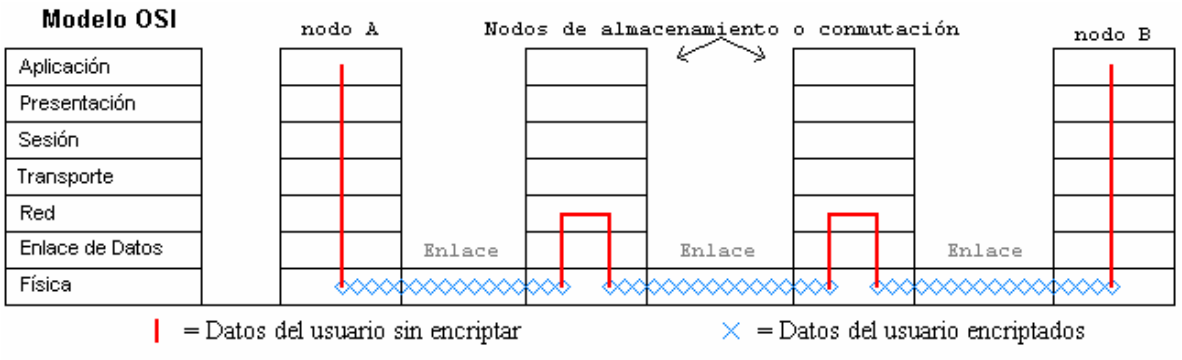


Figura C-1. Encriptación de enlace por enlace

Este tipo de encriptación es muy efectiva. Dado que todo es encriptado, un criptoanalista no puede obtener alguna referencia sobre la estructura de la información que viaja. El enemigo no tiene idea de quien esta hablando a quien, la longitud de los mensajes enviados, el número de veces que ellos se comunican al día, etcétera. Esto es llamado **seguridad en el flujo del tráfico** (“traffic-flow security”): al enemigo no solo carece del acceso a la información, tampoco puede accesar al conocimiento de donde y cuanta información esta fluyendo.

La seguridad no depende de ninguna técnica de manejo de tráfico. La administración de llaves es simple, únicamente los extremos de cada enlace necesitan una llave común, y está puede cambiar independientemente del resto de la red.

En el caso de una línea de comunicación síncrona, después de la inicialización la línea trabaja indefinidamente, recuperando automáticamente bits o errores de sincronización. La línea encripta todo mensaje enviado de un extremo del enlace al otro; y si no hay algún mensaje, sólo cifra y descifra datos aleatorios. Un enemigo a la escucha no tiene idea cuando están siendo enviados mensajes y cuando no; no tienen idea cuando los mensajes inician y terminan. Lo único que ve es un torrente sin fin de bits con apariencia aleatoria.

En el caso de una línea de comunicación asíncrona, la diferencia es que el adversario puede obtener información sobre la razón de transmisión. Si esta información debe ser ocultada, se pueden enviar mensajes de relleno (valor aleatorio) durante los períodos inactivos.

El gran problema con la encriptación en la capa física es que cada enlace físico de la red necesita ser encriptado: dejar cualquier enlace sin encriptación pone en peligro la seguridad de toda la red. Si la red es grande, el costo puede incrementarse rápidamente hasta convertirse en prohibitivo. Además, cada nodo en la red debe ser protegido, dado que procesa información descifrada. Si todos los usuarios de la red son confiables y todos los nodos se encuentran en lugares seguros, esto es tolerable. Pero aún en las compañías más simples, la información debe mantenerse en secreto dentro del mismo departamento. Si la red, accidentalmente, direcciona mal o no reencripta la información y la transmite, cualquiera podrá leerla.

A.2 Encriptación de Extremo a Extremo

Existen dos aproximaciones para obtener una encriptación de extremo a extremo [20]:

- Si el proceso se realiza en las capas intermedias de la arquitectura OSI.
- Si el proceso se realiza en las capas superiores de la arquitectura OSI.

Una opción es colocar el equipo de encriptación en las capas intermedias de la arquitectura OSI, entre la capa de red y la capa de transporte. Este dispositivo debe entender los datos de acuerdo a los protocolos superiores a la capa 3 y encriptar únicamente las unidades que transportan datos, para después ser combinadas con la información de enrutamiento no encriptada y ser enviadas a las capas inferiores para su transmisión.

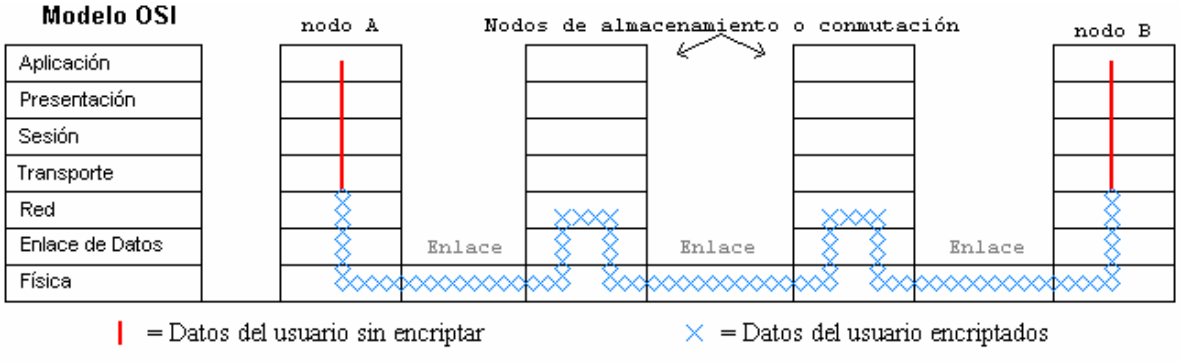


Figura C-2. Encriptación de Extremo a Extremo en las capas intermedias

Esta implementación evita el problema de *descifrar* / *procesar* / *cifrar* en cada nodo de la capa física. Al proveer encriptación de extremo a extremo, los datos se mantienen encriptados hasta alcanzar su destino final. El principal problema con la encriptación de extremo a extremo es que la información de enrutamiento de los datos no es encriptada; un buen criptoanalista puede aprender mucho de quien esta hablando a quien, cuantas veces y por cuanto tiempo; sin conocer el contenido de sus conversaciones. Además, la administración de las llaves es más complicada, dado que se debe asegurar para cada par de usuarios tengan las llaves correctas a usar. Esto implica la existencia de un ente que genere las llaves de sesión o almacene las llaves públicas, ente sujeto a ataques por parte del enemigo.

Construir equipo para encriptación de extremo a extremo es difícil. Cada sistema de comunicación particular tiene sus propios protocolos. Algunas veces las interfaces entre las capas no están bien definidas, haciendo la tarea más complicada.

Si la encriptación se realiza en una de las capas superiores de la arquitectura de comunicación, como en la capa de aplicación o en la de presentación, entonces esta operación se vuelve independiente del tipo de red de comunicación utilizada. Todavía es una encriptación de extremo a extremo, pero la implementación no debe preocuparse en códigos de línea, sincronización entre módems, interfaces físicas, y todo eso. En los antiguos días de la encriptación electromecánica, el cifrado y descifrado se efectuaba fuera de línea; con este acercamiento estamos a un paso de esa antigua implantación.

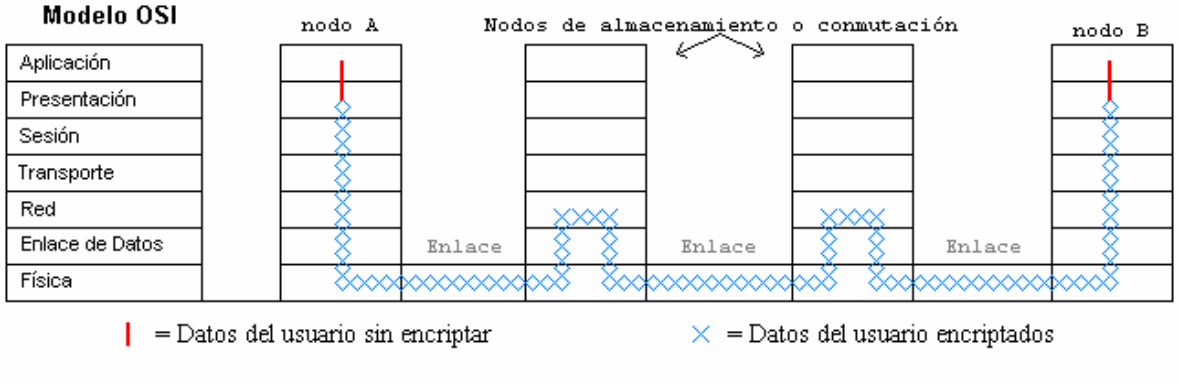


Figura C-3. Encriptación de Extremo a Extremo en las capas superiores

La encriptación en estas capas superiores interactúa con el “software” del usuario. Este “software” es diferente para cada arquitectura particular de computadora, y la encriptación debe ser optimizada para diferentes sistemas de cómputo. La encriptación puede ocurrir en el “software” mismo o en un “hardware” especializado. En el último caso, la computadora pasará los datos al “hardware” especializado para encriptarlo antes de enviarlo a las capas inferiores de la arquitectura de comunicación para transmitirlo. Este proceso requiere de inteligencia y no es factible para terminales tontas. Adicionalmente, pueden existir problemas de compatibilidad con diferentes tipos de computadoras.

La principal desventaja de la encriptación de extremo a extremo es que ésta permite el análisis de tráfico. Este análisis es el análisis de mensajes encriptados: de donde vienen, a donde van, que tan largos son, cuando son enviados, que tan frecuentes son, cuando

estos coinciden con eventos como reuniones, etcétera. Mucha información importante esta dentro de estos datos, y un criptoanalista desea poner sus manos sobre ella.

A.3 Combinando las Dos

Combinando la encriptación de extremo a extremo con la de enlace por enlace, se obtiene la forma más efectiva para asegurar una red, aunque también es la más costosa. La encriptación de cada enlace físico hace imposible cualquier análisis del enrutamiento de información, mientras que la encriptación de extremo a extremo evita la fuga de datos no cifrados en los distintos nodos de la red. La administración de llaves para los dos esquemas puede ser totalmente separada: los administradores de la red pueden encargarse de la encriptación a nivel capa física, mientras que los usuarios tienen la responsabilidad de la encriptación de extremo a extremo.

Anexo B: Factores a Considerar al Comprar por Internet

1. No proporcionar información de su Tarjeta de Crédito, Cuentas de Inversión o Cheques para navegar a través de Internet, a menos que hayas solicitado algún producto o servicio.

2. Verificar que las páginas sean seguras (deberá aparecer un candado cerrado en la esquina inferior).

3. Sólo proporcionar datos confidenciales a sitios que cuenten con prestigio y que ya sean conocidos por usted.

4. Nunca proporcionar a terceros el código de seguridad o password que asignan algunos sitios para efectuar compras.

5. Leer cuidadosamente los términos de uso y garantía.

6. Conservar el folio, comprobante o cualquier dato que respalde la compra.

7. Asegurarse de que el comercio no realice cargos a su cuenta después de que la compra sin que así usted haya solicitado.

8. Cerciorarse de que el sitio comercial cuente con una página o un certificado emitido por un tercero confiable a través del cual usted pueda comprobar la seriedad del negocio. A continuación se muestra un ejemplo:

WWW.SEARS.COM.MX is a VeriSign Secure Site

Security remains the primary concern of on-line consumers. The VeriSign Secure Site Program allows you to learn more about web sites you visit before you submit any confidential information. Please verify that the information below is consistent with the site you are visiting.

Name	WWW.SEARS.COM.MX
Status	**Valid**
Validity Period	31-MAR-06 - 31-MAR-07
Server ID Information	Country = MX State = Distrito Federal Locality = Mexico Organization = Sears Roebuck de Mexico Organizational Unit = Sistemas POS Organizational Unit = Terms of use at www.advantage-security.com/rpa (c) 04 Organizational Unit = Authenticated by Advantage Security Systems Organizational Unit = Member, VeriSign Trust Network

	Common Name = www.sears.com.mx

If the information is correct, you may submit sensitive data (e.g., credit card numbers) to this site with the assurance that:

- This site has a VeriSign Secure Server ID.
- VeriSign has verified the organizational name and that SEARS ROEBUCK DE MEXICO has the proof of right to use it.
- This site legitimately runs under the auspices of SEARS ROEBUCK DE MEXICO.
- All information sent to this site, if in an SSL session, is encrypted, protecting against disclosure to third parties.

To ensure that this is a legitimate VeriSign Secure Site, make sure that:

1. The original URL of the site you are visiting comes from WWW.SEARS.COM.MX.
2. The URL of this page is https://digitalid.verisign.com.
3. The status of the Server ID is Valid.

Figura G-1. Página "Web" de un sitio seguro para comercio electrónico

Anexo C: Modelo de Seguridad en Java

Java fue diseñado para crear applets Java. Los applets Java permiten que el código pueda ser descargado directamente en un navegador *Web*. Esta tecnología fue una de las primeras en convertir un navegador "Web" en la infraestructura que soporta la ejecución de una aplicación cargada desde la "Web". Dicha infraestructura promete un nuevo paradigma en la informática diferente a la computación tradicional enfocada a equipos aislados. En la computación de equipos aislados, las aplicaciones son cargadas y ejecutadas por el usuario y su máquina. Cuando el usuario necesita realizar actualizaciones de la aplicación, primero se deben obtener dichas distribuciones de fuentes como CD o disquetes; para después realizar la actualización. Los applets Java permiten un nuevo paradigma en el cual el código móvil es descargado dinámicamente al navegador "Web" y automáticamente actualizado cada vez que uno revisita el sitio web del cual el código fue descargado.

Al navegar en Internet y acceder a un sitio "Web", el navegador trae una página la cual contiene un applet, el cual automáticamente se ejecuta en la máquina del cliente (navegante). Idealmente, la ejecución de dicho applet no debe afectar el funcionamiento y/o archivos de la máquina cliente (navegante); pero la realidad es otra: existe el **código malicioso** o **maligno**.

Para tener una idea clara de la naturaleza del código maligno, se presentan los siguientes ejemplos:

1. Un applet podría presentar fotografías obscenas, consignas políticas o ideológicas en la pantalla; o reproducir ruidos irritantes por el sistema de sonido del equipo.
2. Al tener acceso al sistema de archivos del navegante, se tiene un atentado contra el mismo si un applet puede: llenar el disco duro con archivos basura; borrar, dañar o encriptar dichos archivos sin el consentimiento del usuario.
3. Y si el código maligno tiene acceso al sistema de archivos y la capacidad de usar los servicios de comunicación, podría leer información personal del usuario (correo electrónico almacenado, contraseñas, documentos) y enviarlos por un flujo o por correo a través de Internet.

Los diseñadores de Java, conscientes de estos problemas, erigieron una serie de barreras y sistemas defensivos para conformar el **modelo de seguridad de Java**.

Un texto que trata a profundidad la seguridad y evolución en Java es [14].

C.1 Arquitectura Criptográfica Java

La **Arquitectura Criptográfica Java** ("Java Cryptography Architecture", JCA) provee una infraestructura para tener una funcionalidad criptográfica básica en la plataforma Java. El alcance de la funcionalidad criptográfica incluye la protección de los datos contra la corrupción usando funciones y algoritmos criptográficos para resguardar la integridad de los datos. Los algoritmos criptográficos para la generación de firmas usados para la identificación de fuentes de datos y código también están incluidos dentro del JCA. Debido a que las llaves y los certificados son una parte esencial para la identificación de las fuentes de datos y códigos, también se incluyen API's para manejar dichos elementos.

La JCA incluida en la plataforma Java apareció por primera vez con Java 1.1. JCA provee las funciones criptográficas básicas para alcanzar los siguientes propósitos:

- Proveer la infraestructura para proteger la integridad de los datos almacenados o transferidos.
- Identificar al ente principal asociado a los datos que se están transfiriendo o se están recuperando de un almacén.
- Proveer la infraestructura para soportar la generación de llaves y certificados usados para identificar las fuentes de datos.
- Proveer una infraestructura a la cual se puedan conectar diferentes algoritmos criptográficos de diferentes proveedores de servicio.

C.2 Extensión Criptográfica de Java

Los términos encriptación y criptografía algunas veces son usados indistintamente. Sin embargo, Sun se adhiere a la definición de que la criptografía provee las funciones de integridad de datos e identificación de fuentes, la cual es implementada por el JCA. Por encriptación se entiende el uso de funciones para encriptar bloques de datos con el fin de añadir confidencialidad hasta que el dato sea desencriptado por el receptor autorizado. La **Extensión Criptográfica de Java** (“Java Cryptography Extension”, JCE) se provee como una extensión de seguridad para realizar las tareas de encriptación.

En general, se puede argumentar lógicamente que la encriptación es un aspecto esencial de cualquier sistema seguro como para que Sun hubiera incluido al JCE dentro de la plataforma y no manejarlo como una extensión. Pero esta realidad se debe a las restricciones de USA en lo referente a la tecnología de encriptación. Si Sun hubiera incluido al JCE como una parte esencial dentro de la plataforma Java, la exportación de este lenguaje fuera de USA sería imposible.

Anexo D: Manejo de Audio en Java

D.1 Introducción al API de Sonido de Java ("Java Sound API")

El API de sonido de Java es una aplicación de bajo nivel que controla las salidas y entradas de sonido, incluyendo tanto audio digitalizado como datos provenientes de la Interfase Digital de Instrumentos Musicales ("Musical Instrument Digital Interface, MIDI"). El API de Sonido de Java provee control explicito sobre las capacidades normalmente requeridas para las entradas y salidas de sonido.

D.2 Muestreo de Audio

Una señal de audio es una señal u onda que puede ser medida y transformada, mediante el uso de micrófonos, de una señal acústica (compresión de aire que viaja por la atmósfera) a una señal eléctrica. Posteriormente, esta señal eléctrica pasa a través de un convertidor analógico-digital y mediante un proceso de muestreo se obtiene su representación digital. El muestreo, se refiere a tomar valores de la señal en determinados intervalos de tiempo. Esto puede observarse en la figura siguiente:

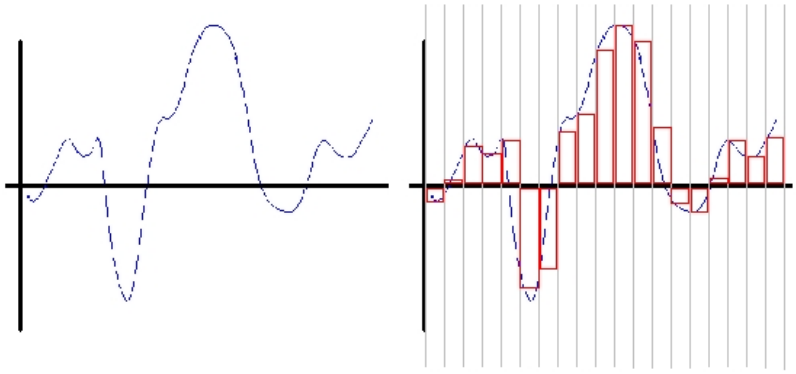


Figura H-1: Señal de Audio y Muestreo

El API de sonido de Java no especifica una configuración de hardware de audio; este ha sido diseñado para permitir que diferentes componentes puedan ser instalados sobre el sistema y accedan a él a través del API.

En este ejemplo, un dispositivo tal como una tarjeta de sonido tiene varios puertos de entrada y de salida, y mezcla todas ellas a través de software. El mezclador puede recibir datos que provengan de un archivo, un flujo de una red, son generados "al vuelo" por un programa de aplicación, o son producidos por un sintetizador de audio.

D.3 ¿Qué es MIDI?

A diferencia de un sonido muestreado, el cual es una representación de dicho sonido en sí mismo, la información de MIDI solo especifica como crear un sonido, especialmente sonidos musicales. La información MIDI no describe sonidos directamente; de hecho, sólo

describe eventos que afectan al sintetizador de sonido. De esta forma, MIDI se comporta como un enorme piano en el que, a través de las teclas, los pedales, “switches” y demás controles, genera sonidos.

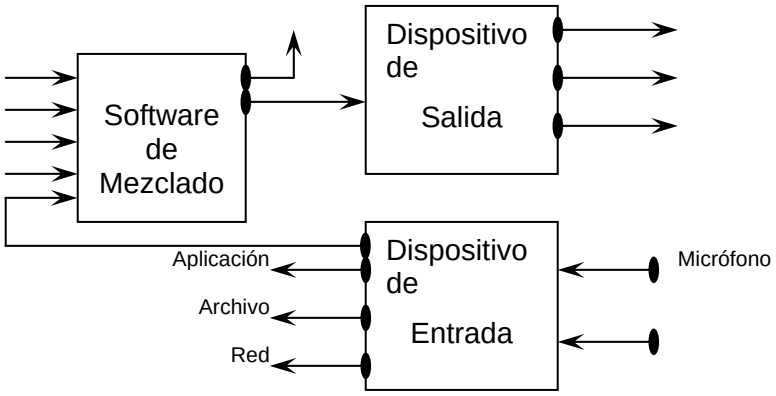


Figura H-2: Arquitectura de Audio Típica

D.4 Interfases para Proveedores de Servicios

Además de los paquetes propios de Java para el manejo de archivos y la configuración de sonido del sistema (javax.sound.sampled y javax.sound.midi), existen dos paquetes más (javax.sound.sampled.spi y javax.sound.midi.spi) que permiten a los desarrolladores crear nuevos recursos para audio común y MIDI.

La implementación del API de sonido de Java permite manejar los servicios básicos de sonido, a través de la Interfase para Proveedores de Servicio (SPI) pueden ser creados servicios adicionales más sofisticados.

D.5 Revisión General del Paquete “javax.sound.sampled”

Este paquete centra su trabajo en el transporte de los datos de audio. Su tarea principal será cómo mover los bytes de audio formateado hacia y fuera del sistema. Estas tareas involucran: abrir y cerrar archivos y administrar “buffers” (almacenes temporales) para producir sonido en tiempo real.

“Streaming” (flujo) es la palabra con la que nos referimos a un movimiento constante de bytes que generan audio en tiempo real. En otras palabras, un flujo de audio es simplemente un conjunto continuo de información de sonido que llega más o menos a la misma velocidad a la que se almacena o reproduce. En el modelo de flujos, particularmente para el caso del audio, usted no necesita saber de antemano que tan grande es el archivo de sonido y cuando terminará este de llegar. Simplemente se requiere de un “buffer” de audio que almacene temporalmente los datos hasta su reproducción o almacenamiento definitivo.

Por otro lado, cuando los datos son reproducidos desde un archivo, el API de sonido de Java permite reproducir el sonido sin necesidad de crear un ”buffer”. Esto se debe a que

el programa asume que se tiene todo el archivo de datos necesario y además, dicho archivo no es tan grande como para saturar la memoria. En este caso, todo el archivo de sonido puede ser precargado completo en la memoria para su reproducción tantas veces como se desee. Por esta misma razón, al momento de reproducir un archivo que se encuentra almacenado, su reproducción se efectúa de forma casi instantánea.

D.6 Revisión General del Paquete "javax.sound.midi"

El estándar MIDI define un protocolo de comunicación para dispositivos de música electrónica, tales como teclados electrónicos y computadoras personales. Los datos MIDI pueden ser transmitidos sobre cables especiales durante conciertos en vivo o pueden ser almacenados, bajo un estándar especial, en archivos que posteriormente serán reproducidos y/o editados.

MIDI es tanto una especificación de "hardware" como de "software". Para entender el diseño de MIDI debemos entender que este fue diseñado pensando en reproducir eventos que suceden en instrumentos electrónicos (tal como pulsar una tecla) o por instrucciones enviadas desde una PC al sintetizador de audio.

Los dispositivos de "hardware" conocen o tienen almacenadas secuencias de notas que son reproducidas por el sintetizador, permitiendo al músico o compositor "tocar" determinada melodía en vivo. Posteriormente, fueron desarrolladas interfases que permiten conectar los instrumentos musicales con el puerto serial de una computadora; de esta forma, se pueden manipular computacionalmente las entradas de este puerto. Hoy en día sin embargo, mediante la incorporación de tarjetas de audio que tienen integradas microcircuitos capaces de efectuar todo el trabajo del sintetizador MIDI, muchos usuarios se limitan únicamente a trabajar con el sintetizador de música digital.

La especificación de MIDI destinada a los elementos de hardware, detalla el uso y función de cada uno de los "pines" para los cables de las interfases MIDI. Por su parte, la porción de la especificación que explica los lineamientos generales que debe seguir la implementación en software, se concentra en la estructura de datos y como los dispositivos sintetizadores deben responder a ellos, es decir, como deben reproducirlos. Es importante entender que los datos MIDI pueden ser secuenciados o moverse en flujo.

GLOSARIO

Glosario de Términos

El siguiente Glosario es para proporcionar información de referencia rápida, sobre la terminología de comunicaciones utilizada con más frecuencia en esta tesis.

- A -

Active Server: Una colección de tecnologías de servidor que se entregan con Windows NT. Estas tecnologías proporcionan un modelo de componentes y secuencias de comandos coherente de servidor, así como un conjunto integrado de servicios del sistema para administración de las aplicaciones componentes, acceso a bases de datos, transacciones y mensajería.

Active X: Conjunto de tecnologías de Microsoft las cuales permiten interactividad con el material existente en el WWW.

Address Resolution Protocol (ARP): Protocolo de Resolución de Direcciones. Es el protocolo de Internet utilizado para crear un mapa dinámico de las direcciones en las áreas de red locales. Permite la conversión de una dirección Internet en una dirección numérica.

Administrador de transacciones: Un servicio del sistema responsable de coordinar el resultado de las transacciones con el fin de conseguir atomicidad. El administrador de transacciones asegura que los administradores de recursos toman decisiones coherentes sobre si la transacción debe realizarse o no.

ADO (ActiveX Data Objects): Objetos ActiveX para Datos. Un conjunto de interfaces de acceso a datos, basadas en objetos y optimizadas para las aplicaciones enfocadas a Internet y centradas en manejo de datos. ADO está basado en una especificación publicada y se incluye con Microsoft Internet Information Server y con Microsoft Visual InterDev.

Advanced Research Projects Agency Network (ARPANET): Red pionera de Internet fundada por ARPA -una agencia gubernamental norteamericana del Departamento de Defensa- en 1969. ARPANET conectaba ordenadores que intercambiaban paquetes de información mediante líneas en “leasing” y sus investigaciones en redes sirvieron de base para el actual sistema.

Algoritmo: Regla o proceso a seguir para realizar una tarea o llegar a la solución de un problema.

Amenaza: Circunstancia o evento que puede causar una denegación de servicio o una destrucción, revelación o modificación de datos no autorizada.

Ancho de Banda (bandwidth): Rango de frecuencias asignadas a un canal de transmisión. Es la capacidad de transporte de datos que se puede enviar a través de una

conexión. Normalmente se mide en megabytes por segundo (MB/s) o en gigabytes por segundo (GB/s).

ANSI (American National Standards Institute): Instituto Nacional Americano de Normas. Organización encargada de aprobar las normas con que se rigen diferentes sectores en los Estados Unidos -incluyendo ordenadores y comunicaciones- y en la que se agrupan asociaciones profesionales, compañías y asociaciones gremiales; es miembro de la International Organization for Standardization (ISO).

Applet: Programa o aplicación de pequeño tamaño – comúnmente programado en el lenguaje Java de Sun Microsystems.

API (Application Program Interfaces): Interfaces de Programación de Aplicaciones. Un conjunto de rutinas que un programa de aplicación utiliza para solicitar y efectuar servicios de nivel inferior ejecutados por el sistema operativo de un equipo. También es un conjunto de convenciones de llamada en programación que definen cómo se debe invocar un servicio a través de la aplicación.

Arquitectura Cliente-Servidor: Un modelo de computación mediante el que las aplicaciones cliente que se ejecutan en un escritorio o en un equipo personal tienen acceso a la información contenida en servidores remotos o en equipos "host". La parte cliente de la aplicación suele estar optimizada para la interacción con el usuario, mientras que la parte servidor proporciona la funcionalidad centralizada multiusuario.

Asociación Mexicana de Estándares para el Comercio Electrónico (AMECE): Organismo de reciente creación cuya misión es la de normalizar las reglas y formatos a utilizar en el comercio electrónico. Interactúa con organismos públicos nacionales, en especial la SHCP de México, así como con organismos internacionales e instituciones privadas.

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

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

Autorización: En lo referente a equipos, especialmente a equipos remotos de una red que están disponibles para más de una persona, el permiso concedido a un individuo para usar el sistema y los datos almacenados en él. La autorización la establece normalmente un administrador del sistema y la comprueba y acepta el equipo. Esto requiere que el usuario proporcione algún tipo de identificación, como un código o una contraseña, que el equipo pueda comprobar con sus registros internos. Los términos permiso y privilegio son sinónimos de autorización.

- B -

Bit (bit/binary digit): Dígito binario. Es la menor unidad de información, con valores posibles 1 y 0.

BPS (bps): Acrónimo de bits por segundo. Es la medida estándar de la velocidad de transmisión de datos.

Bug: Fallo de diseño o seguridad en un programa o equipo.

Byte (byte): Un conjunto de bits tratados como una unidad. Normalmente tiene una longitud de 8 bits (octeto). La capacidad de almacenamiento de un dispositivo, frecuentemente

- C -

Certificado Digital: Un archivo, obtenido de una entidad emisora de certificados, que se utiliza para comprobar el origen de los datos enviados a través de una red; también se denomina certificado de autentificación.

Cifrado: Transformación de un mensaje en otro, utilizando una clave para impedir que el mensaje transformado pueda ser interpretado por aquellos que no conocen la clave. Véase la sección 1.2 del presente documento donde se da un análisis más detallado del uso de este término.

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

Clave de sesión / Clave de cifrado: Clave utilizada en algoritmos simétricos para cifrar y descifrar los mensajes en una única sesión.

Clave Privada: Clave personal que no es conocida por el resto de los usuarios y que es utilizada para crear firmas digitales y, dependiendo del algoritmo, para descifrar mensajes cifrados con la correspondiente clave pública.

Clave Pública: Clave de usuario que es conocida por el resto de los usuarios y que es utilizada para verificar firmas. Dependiendo del algoritmo, se usa para cifrar mensajes que pueden ser descifrados con su correspondiente clave privada.

Clave Simétrica: Clave única usada en los algoritmos simétricos tanto para cifrar como para descifrar un mensaje.

Código de bytes: El formato ejecutable de código Java que se ejecuta en la Máquina Virtual de Java (“Java Virtual Machine”, JVM) Java. También se denomina código interpretado, pseudo código, “byte code” y p-code.

Concentrador (HUB): Punto de conexión común para dispositivos dentro de una red, normalmente unen a segmentos de una red. El hub se encarga de distribuir la información recibida por cualquiera de sus puertos a todos los demás.

Contraseña (password): Palabra o cadena de caracteres, normalmente secreta, para acceder a través de una barrera. Se usa como herramienta de seguridad para identificar usuarios de una aplicación, archivo, o red. Puede tener la forma de una palabra o frase de carácter alfanumérico, y se usa para prevenir accesos no autorizados a información confidencial.

Cookie: Fichero de texto instalado en el directorio del navegador, en el que se guarda información sobre preferencias del usuario y datos diversos, que activan ciertas respuestas por parte de otros sistemas a los que se conecta.

Control de acceso: Los elementos e instrumentos de salvaguarda necesarios para garantizar a los usuarios la seguridad de los datos y demás activos del sistema de comunicación y sus aplicaciones.

Correo Electrónico (E-mail): Un sistema mediante el cual un usuario de un equipo puede intercambiar mensajes con otros usuarios (o grupos de usuarios) por medio de una red de comunicaciones. El correo electrónico es una de las aplicaciones más populares de Internet.

Cracker: Intruso; individuo que intenta penetrar en un ordenador o sistema informático, ilegalmente y generalmente con intenciones malsanas -a menudo confundido con hacker.

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

- D -

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

Datos personales: Cualquier información referente a una persona identificada,

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

DES*:* Data Encryption Standard. Algoritmo de cifrado / descifrado, diseñado y reglamentado en EEUU.

Descifrado*:* Función inversa al cifrado.

DHTML: HTML dinámico. Un conjunto de innovadoras características presentes en Internet Explorer versión 4.0 que pueden usarse para crear documentos HTML que cambian su contenido dinámicamente e interactúan con el usuario. Al usar DHTML, los autores pueden aportar a las páginas “Web” efectos especiales sin depender de programas del servidor.

- E -

EDI (Electronic Data Interchange): Intercambio Electrónico de Datos. Protocolo creado a principios de los años 70 para permitir, a las grandes compañías, la transmisión de información a través de sus redes privadas; y el cual se ha tratado de adaptar a los actuales sitios “Web” corporativos.

Encriptación: Acción de proteger la información mediante técnicas criptográficas ante modificaciones o utilización no autorizada. Véase la sección 1.2 del presente documento donde se da un análisis más detallado del uso de este término.

Entidad Emisora de Certificados (Certificate Authorithy): Una entidad que emite, administra y revoca certificados.

- F -

FAQ (Frequently Asked Questions): Preguntas Frecuentemente Preguntas. Lista de preguntas que se efectúan con gran frecuencia, y sus respuestas. Existen cientos de FAQs sobre los temas más diversos y su objetivo es evitar un aluvión de preguntas obvias - sobre todo a los "newsgroups".

File Transfer Protocol (FTP): Protocolo para la Transferencia de Archivos. Protocolo que permite enviar y recibir ficheros - de un ordenador a otro - dentro de Internet; hay miles de sitios en la red que ofrecen ficheros y programas de todo tipo de forma desinteresada.

Firewall: Traducido literalmente *muro de fuego*; conceptualmente significa *muro corta-fuego* en la lengua inglesa. Se trata de un programa y/o equipo que protege a una red de otra red. En su concepción más simple, el "firewall"permite que una máquina acceda a Internet desde una red local, pero impide que las otras máquinas fuera de esa red accedan a la máquina.

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

- G -

Gateway: Pasarela, dispositivo, ordenador o programa que conecta redes - que normalmente serían incompatibles - permitiendo el intercambio de información entre ellas; también llamado "router".

- H -

Hacker: Experto en los entresijos de programas, ordenadores, sistemas, redes en general e Internet en particular. En lenguaje de la Red, el "hacker" es un personaje no perjudicial quien no debe ser confundido con el "cracker".

HTML (HyperText Markup Language): Lenguaje de Marcaje de Hiper Texto. Lenguaje utilizado en el WWW para crear páginas "Web" que se conectan con otros documentos - se trata de códigos que dictan el formato y composición de la página, estructurándola de forma que sea accesible y creando enlaces con otras páginas o ficheros de la red.

HTTP (HiperText Transfer Protocol): Protocolo para transferir ficheros o documentos en el WWW.

- I -

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

Interfaz: Un grupo de operaciones o métodos relacionados lógicamente que proporciona acceso a un objeto componente.

International Organization for Standardization (ISO): Organización Internacional para la Normalización. Organización fundada en 1946 responsable de establecer normas internacionales en diversas áreas - incluyendo comunicación e información.

Internet (Internet): Serie de redes locales, regionales, nacionales e internacionales interconectadas, unidas usando TCP/IP. Internet une muchos gobiernos, universidades y centros de investigación. Proporciona E-mail, login remotos y servicios de transferencia de archivos.

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

Intranet: Red de comunicación interna o privada que utilizan empresas u organizaciones, diseñada en base a los protocolos de Internet y que puede estar o no conectada a Internet.

IP (Internet Protocol): Protocolo de Interred. Protocolo estándar utilizando por los sistemas que se comunican por el Internet.

ISAPI (Internet Server Application Program Interfaces): Interfaz de Programación de Aplicaciones de Servidor Internet. Una interfaz de programación de aplicación que reside en un equipo servidor para el inicio de los servicios de software ajustados para el sistema operativo Microsoft Windows NT. Es una API para desarrollar extensiones para Microsoft Internet Information Server y otros servidores HTTP compatibles con la interfaz ISAPI.

ISP (Internet Services Provider): Proveedor de Servicios de Internet. Es una organización, una compañía o una institución docente, que permite a los usuarios remotos tener acceso a Internet proporcionándoles conexiones de acceso telefónico o mediante la instalación de líneas dedicadas.

- J -

Java: Lenguaje de programación desarrollado por Sun Microsystems con un código compatible con todas las plataformas de ordenadores, por lo que puede ser recibido a través de Internet y utilizado de inmediato sin temor a virus o daño a los ficheros.

JavaScript: Un lenguaje de secuencias de comandos que evolucionó a partir del lenguaje “LiveScript” de Netscape y que se hizo más compatible con Java. Utiliza una página HTML como interfaz.

- K -

Kerberos: La base de la mayoría de los servicios de seguridad del Entorno de Computación Distribuida (Distribuited Computing Enviroment, DCE). Kerberos proporciona un uso seguro de los componentes de software distribuidos.

- L -

Login (login): Nombre que se usa para acceder a un sistema de ordenadores. No es secreta, si lo fuera sería una password (clave). Acción de entrar en un sistema de ordenadores.

- Ll -

Llave (Key): Frase o conjunto de caracteres que permiten descodificar un mensaje cifrado.

- M -

Máquina Virtual Java: El mecanismo que el lenguaje Java utiliza para ejecutar el código de bytes de Java en un equipo físico. La máquina virtual convierte el código de bytes a la instrucción nativa del equipo de destino.

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

Modelo OSI (modelo de Interconexión de sistemas abiertos): El modelo de referencia OSI proporciona la base para el desarrollo de estándares relativos a las redes. Este modelo enumera siete capas que definen las actividades que deben tener lugar cuando se comunican los dispositivos a través de una red. Estas siete capas (de arriba a abajo) son: aplicación, presentación, sesión, transporte, red, enlace y física.

- N -

Navegador *(Browser):* Visualizador; programa o aplicación para navegar a través del Web (WWW), tal como Netscape o Internet Explorer. Permite al usuario acceder a documentos, imágenes y ficheros localizados en servidores “Web”.

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

- O -

Objeto: Es la unidad básica de la programación orientada a objetos, la cual comprende rutinas y datos, y es tratada como una entidad discreta. Un objeto se basa en un modelo específico, donde un cliente que utiliza los servicios de un objeto obtiene acceso a los datos del objeto a través de una interfaz que consta de un conjunto de métodos o

funciones relacionados. El cliente puede llamar después a estos métodos para realizar operaciones.

ODBC (Open DataBase Connectivity): Conectividad Abierta a Bases de Datos. Una interfaz de programación de aplicaciones que permite a las aplicaciones tener acceso a datos desde diversas especificaciones estándar de orígenes de datos para acceso a bases de datos multiplataforma.

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

OSI (Open Systems Interconnection): Interconexión de Sistemas Abiertos. Arquitectura modular para red desarrollada por la ISO, la cual usa siete capas para soportar comunicaciones abiertas entre equipos de diferentes fabricantes.

- P -

Página "Web": Un documento de la WWW. Las páginas pueden contener prácticamente cualquier cosa, por ejemplo noticias, imágenes, películas y sonidos.

Páginas de Servidor Activo (Active Server Pages, ASP): Un entorno de secuencias de comandos de servidor que ejecuta secuencias de comandos ActiveX y componentes ActiveX en un servidor. Los programadores pueden combinar secuencias de comandos y componentes para crear aplicaciones basadas en "Web".

PGP (Pretty Good Privacy): Privacidad Bastante Buena. Programa de libre distribución, escrito por Phil Zimermann; el cual impide mediante técnicas de criptografía, que ficheros y mensajes de correo electrónico puedan ser interpretados por personas no autorizadas. También puede utilizarse para firmar electrónicamente un documento o un mensaje, realizando así la autentificación del autor.

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

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

- Q -

Query: Consulta. Mensaje solicitando el valor de una variable o conjunto de variables.

- R -

RSA: Rivest-Shamir-Adleman. Algoritmo criptográfico de cifrado de clave asimétrica, utiliza una clave para cifrar y otra para descifrar.

- S -

SDK (Software Development Kit): Conjunto de herramientas para el desarrollo de software. Herramientas de una compañía específica, utilizadas por los programadores para crear nuevas aplicaciones.

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

Servidor "Web" seguro: Servidor "Web" que utiliza protocolos de seguridad (SSL, S-HTTP o PCT) al ejecutar transacciones con él. Un protocolo de seguridad utiliza técnicas de cifrado y autenticación como medios para incrementar la confidencialidad y la fiabilidad de las transacciones.

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

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

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

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

- T -

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

Telnet: Protocolo de comunicaciones estándar que conecta un ordenador con Internet, convirtiéndolo en una terminal del sistema.

- U -

URL: Acrónimo de Universe Resource Locator. Este término hace referencia a una dirección *Web*.

- W -

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

Worm (Gusano): También conocido como “Great Worm”, fue introducido por Robert T. Morris en Internet en noviembre de 1998 y se propagó de tal manera que colapsó más de seis mil sistemas.

- X -

X509: Norma estándar que define un entorno de autentificación y seguridad. Forma parte de la norma X.500 de UIT -T.

BIBLIOGRAFÍA Y REFERENCIAS

Bibliografía y Referencias

Libros

1 **Aldegani, Gustavo Miguel**
Seguridad Informática
MP Ediciones, Argentina, 1997
1ª. Edición

2 **B.Schneier,**
Applied Cryptography,
Ed. Wiley, EE.UU.,1996
2ª. Edición

3 **Baker, David**
Java Expert Solutions
QUE Corporation, EE.UU., 1997
1ª. Edición

4 **Bobadilla Sancho, Jesús**
HTML Dinámico, ASP y JavaScript
Ed. Alfaomega, España, 2000
1ª. Edición

5 **Carballar Falcon, José Antonio**
Redes, Aplicaciones y Costes
Ed. RA-MA, España, 1993
1ª. Edición

6 **Cebrian Ruz, Antonio**
Guía Práctica de Comunicaciones y Redes Locales
Colección de Informática de Gestión
Ed. Gustavo Gili

7 **Cohan, Peter**
El negocio está en Internet
Pearson Educación, México, 2000
1ª. Edición

8 **Cooper, Frederic**
Implementing Internet Security
New Riders Publishing, E.E.U.U., 1995
1ª. Edición

9 **Danesh, Arman**
Tatters, Wes
JavaScript 1.1 Developer's Guide
Editorial Prentice Hall, E.E.U.U., 1997
1ª. Edición

10 **Dertouzos, Michael**
El Nuevo Mundo de la Informática
Editorial Planeta, México, 1997
1ª. Edición

11 **Gates, Hill**
Los Negocios en la era digital
Plaza & Janes Editores, México,1999
1ª. Edición

12 **Halsall, Fred**
Comunicación de datos, redes de computadoras y sistemas abiertos
Pearson educación, México, 1998
4ª. Edición

13 **Hopson, K.C.**
Ingram, Stephen
Developing Professional Java Applets
SAMS Publishing, E.E.U.U., 1996
1ª. Edición

14 **Jaworski, Jaime**
Perrone, Paul
Java Security Handbook
SAMS Publishing, E.E.U.U., 2000
1ª. Edición

15 **Lucena, Manuel J.**
Criptografía y Seguridad en Computadoras
4ta Edición V 0.62

16 **Menezes, Alfred**
Handbook of Applied Cryptography
Editorial CRC Press, EE.UU., 1997
4ª. Edición

17 **Moreno Navarrete, M. Ángel**
DERECHO-e Derecho del Comercio Electrónico
Ediciones Jurídicas S.A., Marcial Pons España, 2002
1ª. Edición

18 **Real Academia Española**
Diccionario de la Lengua Española.
Editorial Espasa Calpe S.A., España, 2001
22ª. Edición

19 **Roldán Sauma, Marcelo**
Knorr Jorlene Marie
La Protección del Consumidor en el Comercio Electrónico.
San José: Investigaciones Jurídicas S.A., 2001
1ª. Edición

20 **Schneier, Bruce**
Applied Cryptography (protocols, algorithms and source code in C)
Editorial John Wiley &Sons, EE.UU., 1996
2ª. Edición

21 **Tanenbaum, Andrew S.**
Redes de Computadoras
Editorial Prentice Hall, México, 1997
3ª. Edición

22 **Walnum, Clayton**
Java by Example
QUE Corporation, EE.UU., 1996
1ª. Edición

23 **Walther, Stephen**
Levine, Jonathan
Aprendiendo Programación para E-Commerce con ASP en 21 días
Pearson Educación, México, 2000
2ª. Edición

Tesis

24 **Borghello, Cristian F.**
Seguridad Informática: Sus Implicancias e Implementación
2001

25 **Aldama Ramírez, Gabriel**
Mares Canales, José Francisco
Encriptado de Información en Redes de Datos para el Comercio Electrónico
2001

Ensayos

26 **Ellison, Carl y Schneier, Bruce**
Ten Risks of PKI: What you're not being told about Public Key Infraestructure
2000

27 **Schneier, Bruce**
Security in the Real World: How to evaluate Security Technology
1999

28 **Schneier, Bruce**
Cryptography Design Vulnerabilities
1998

29 **Anderson, Ross; Needham, Roger**
Robustness principles for public key protocols
1996

30 **Blaze, MattDiffie; Whitfield, Rivest; Ronald, Schneier, Bruce; Shimomura, Tsutomu; Thompson, Eric; Wiener, Michael**
Minimal key lengths for symmetric ciphers to provide adequate commercial security
2002

*Estos ensayos pueden ser descargados de la biblioteca digital de Counterpane

Páginas Web

31	Algoritmos Criptográficos	http://www.scramdisk.clara.net/pgpfaq_sp.html#REFSch96a
32	ASP, SDK y Personal Web Server de Microsoft	http://www.microsoft.com
33	Calidad de la Información	http://exlibris.usal.es/merlo/escritos/calidad.htm
34	Certificados Digitales	http://www2.sharesafe.net/sharesafe/TuturialPKI6.asp
35	Code Signing for Java Applets	http://www.suitable.com/Doc_CodeSigning.shtml
36	Comercio Electrónico	http://www.eumed.net/ce/2005/orc-ce.htm#_ftn4
37	Cryptix	http://www.cryptix.org
38	Criptografía	http://www.dat.etsit.upm.es/~mmonjas/cripto/01.html
39	Funciones Resumen	http://dat.etsit.upm.es/~mmonjas/cripto/04.html
40	JDK 1.3	http://www.java.sun.com
41	Organización Mundial de Comercio	http://www.wto.org/spanish/thewto_s/whatis_s/tif_s/bey4_s.htm

42	Seguridad en Aplicaciones Java	http://www.instisec.com/publico/vercurso.asp?id=11#cap7
43	Seguridad en el Comercio Electrónico	http://www.iec.csic.es/criptonomicon/comercio/
44	Seguridad Física	http://es.tldp.org/Manuales-LuCAS/doc-como-seguridad-fisica/COMO-seguridad-fisica.html
45	Seguridad en Redes	http://www.map.es/csi/silice/Seg1.html
46	S-HTTP	http://www.iec.csic.es/criptonomicon/shttp.html

Ayudas en línea

47 Documentación HTML del API del CryptixJCE

48 Documentación HTML del API del Cryptix3.x.x

49 Documentación HTML del API del JCE versiones Beta, EA y EA2, creadas por Sun Microsystems

50 Documentación HTML del API del JCE creado por Bouncy Castle

51 WinHelp para Sun-JDK 1.3.1 de Franck Allimant

52 WinHelp SDKDOCS de Microsoft JDK 4.0

53 WinHelp JAVADOCS de Microsoft JDK 4.0

54 WinHelp INTEGRATION de Microsoft JDK 4.0