UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGON

ALGORITMO ´DES´ APLICADO A LA SEGURIDAD DE BASES DE DATOS MULTIDISCIPLINARIAS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

VERÓNICA CARMONA LEÓN

DIRECTOR DE TESIS:

M.I. Israel Nava Bravo

FES ARAGON MÈXICO 2010

AGRADECIMIENTOS

Primero que nada agradezco a dios por dejarme llegar a esta etapa de mi vida, con bastante camino recorrido lleno de logros personales, deportivos y académicos pero lo principal por dejarme vivirlos y hacer realidad todos mis sueños.

A mis padres que con mucho sacrificio, amor y esfuerzo me educaron para poder ser una persona de bien y sobre todo por su apoyo infinito para poder ser una profesionista.

A mis hermanos que con su apoyo incondicional nunca me dejaron ni dejaran caer en todos los momentos de mi vida.

A mi familia y amigos que siempre están ahí brindándome su apoyo en los momentos que más lo necesitaba.

A mis abuelos Rodolfo, Dolores y Socorro que aunque están lejos siempre me brindaron su confianza desde pequeña y que gracias a eso siempre los llevó en mi memoria.
Y a mi abuelito Hermenegildo León que continúa con nosotros y que espero lo haga por muchos años más, esto es en su honor para que vea que su nieta logra lo que se propone.

TEMARIO

INTRODUCCIÓN

La razón de la implementación de un algoritmo de encriptación es para aumentar la seguridad en las bases de datos, este tema es para tratar de satisfacer las necesidades de las empresas, ya que en la actualidad existen muchas anomalías en el manejo de la información.

El robo de datos, el mal uso de ellos, las modificaciones externas para cambiar la información, son motivos de los fracasos de proyectos importantes de ciertas empresas en la actualidad, es por eso que deseamos proteger la información mas importante de las organizaciones, así que el tema es para proponer un algoritmo de encriptación llamado algoritmo 'DES' (Data Encryption Standard) que nos ayude a mejorar los sistemas de seguridad de las empresas.

La información es uno de los activos más importantes de las entidades, y de manera especial en algunos sectores de actividad. Es indudable que cada día las entidades dependen de mayor medida de la información y de la tecnología, y que los sistemas de información están más soportados por la tecnología, frente a la realidad de hace pocas décadas.

Hace algunos años la seguridad era más fácil con arquitecturas centralizadas y terminales no inteligentes, pero hoy en día los entornos son muy complejos, con diversidad de plataformas y proliferación de redes internas y externas, incluso con enlaces internacionales.

La explosión tecnológica facilita las labores cotidianas, sin embrago, se deben tomar nuevas medidas en cuanto a la implementación de los sistemas de cómputo para que las personas no puedan acceder a información que no les pertenece.

En virtud que todavía no existen procedimientos legales para ser aplicados contra criminales informáticos, es una obligación individual protegerse tanto de los atacantes externos como internos; para estos es importante tener una base de conocimiento para comprender mejor que es lo que está pasando, y saber como profundizar cada vez más y en mejor forma el tema.

Uno de los problemas más graves es el de la seguridad. Los países industrializados y los que no lo son aún completamente, día a día se vuelven más dependientes de la tecnología informática, de las computadoras y de la red mundial que se conecta. Esta dependencia se ha convertido en una amenaza al bienestar económico, a la seguridad ciudadana y a la seguridad nacional de muchos países, aún de los más poderosos. Cabe señalar que las amenazas van en aumento, y que por lo tanto, se demanda gente capacitada y de investigación de nuevas tecnologías.

La seguridad de los sistemas de información (SSI) está relacionada con las disponibilidad, confidencialidad e integridad de la información tratada por las computadoras y las redes de comunicación.

Los sistemas de información son considerados parte fundamental de una organización, y se debe otorgar la misma importancia a la información que estos procesan y almacenan.

El desarrollo de las Bases de Datos colectivas es sin duda una de las actividades más importantes en el campo de la informática, actualmente es impresionante observar como crecen en importancia y en volumen los archivos de datos de las computadoras. Las tasas de crecimiento de la capacidad de almacenamiento de las computadoras, ha hecho posible que se tenga un desmesurado crecimiento de la disponibilidad de información, con lo que se apoya definitivamente el gran desarrollo industrial, comercial, científico y de servicios de nuestro tiempo; cuanto mayor es la cantidad de datos a que tiene acceso la computadora, tanto mayor es su potencial. Es por esto que las bases de datos cobran demasiada importancia en la actualidad.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información.

Con la infraestructura de comunicaciones que existe actualmente, las redes de cómputo han alcanzado gran auge, con el advenimiento de internet se ha abierto a los usuarios posibilidades nunca imaginadas. Actualmente una persona puede tener acceso a información localizada físicamente en otro lugar, incluso al otro lado del mundo, sin siquiera moverse de su lugar. Por otro lado a pesar que la mayor parte de las empresas prefieren abstenerse de dar parte cuando se suscita un incidente de seguridad, el índice de delitos informáticos realizado por medio de redes crece exponencialmente.

El objetivo final de esta implementación es llegar a proteger más los sistemas de información manejados en las bases de datos, con el fin de satisfacer las necesidades de las empresas. Y el buen manejo y uso de la información.

Es por eso que el presente trabajo se abordara con estos temas por medio de la siguiente organización:

⇨ En el capitulo uno, se verán los antecedentes de las bases de datos; así como los conceptos básicos relacionados y sus fundamentos.

⇨ En el capitulo dos, se revisará los términos generales de la seguridad en informática; antecedente y conceptos básicos.

⇨ En le capitulo tres se tratará el tema de la criptografía; su historia, conceptos básicos y la importancia de esta y los sistemas de cifrado dentro de la seguridad informática.

⇨ En el capitulo número cuatro, se revisara la metodología del Algoritmo “DES”; su descripción paso a paso, la encriptación y desencriptación del mismo.

⇨ En el quinto capitulo, se realizará la implementación de nuestro algoritmo con sus pruebas y resultados.

OBJETIVOS:

- Ayudar a crear una conciencia de la importancia de la seguridad en las bases de datos
- Conocer los tipos de seguridad informática aplicadas a las bases de datos y crear conciencia de las misma
- Poner de manifiesto la necesidad y justificación de protección de la información, tanto almacenada como transmitida
- Que el lector adquiera un grado de análisis y pueda planificar una política de seguridad, analizando los riesgos y conociendo las medidas que debe aplicar para obtener una solución
- Introducir al lector las técnicas de elaboración del algoritmo y procedimientos para su implementación en las bases de datos.

Como resultado de este trabajo y de acuerdo a lo planteado, se deberá estar en la posibilidad de llegar a conclusiones propias acerca de la importancia de la seguridad en las bases de datos dentro de cada una de las diversas organizaciones o empresas que deseen tal implementación.

CAPITULO 1. BASES DE DATOS

1.1 ANTECEDENTES

El procesamiento de datos impulsa el crecimiento de los computadores, como ocurriera en los primeros días de los computadores comerciales. De hecho la automatización de las tareas de procesamiento de datos precede a los computadores. Las tarjetas perforadas, inventadas por Holierith, se usaron en los principios del siglo XX para registrar los datos del censo de los EE.UU., y se usaron sistemas mecánicos para procesar las tarjetas y tabular los resultados. Las tarjetas perforadas posteriormente se usaron ampliamente para introducir datos en los computadores.

-Década de 1950 y principios de la década de 1960

Se desarrollaron las cintas magnéticas para el almacenamiento de datos. Las tareas de procesamiento de datos tales como las nóminas fueron automatizadas, con los datos almacenados en cintas. El procesamiento de datos consistía en leer datos de una o más cintas y escribir datos en una nueva cinta. Los datos también se podían introducir desde paquetes de tarjetas perforadas e impresos en impresoras. Por ejemplo, los aumentos de sueldo se procesaban introduciendo los aumentos en las tarjetas perforadas y leyendo el paquete de cintas perforadas en sincronización con una cinta que contenía los detalles maestros de los salarios. Los registros debían estar igualmente ordenados. Los aumentos de sueldo tenia que añadirse a los sueldos leídos de la cinta maestra y escribirse en una nueva cinta; esta nueva cinta se convertía en una cinta maestra.
Las cintas y los paquetes de tarjetas perforadas solo podían leerse secuencialmente, y los tamaños de datos eran mucho mayores que la memoria principal: así, los programas de procesamiento de datos tenia que procesar los datos según un determinado orden, leyendo y mezclando datos de cintas de tarjetas perforadas.

- Finales de la década de 1960 y la década de 1970

El amplio uso de los discos fijos a finales de la década de 1960 cambió en gran medida el escenario del procesamiento de datos, ya que los discos fijos permitieron el acceso directo a los datos. La ubicación de los datos en los discos no era importante, ya que a cualquier posición del disco se podía acceder en solo decenas de milisegundos. Los datos se liberaron de la tiranía de la secuencialidad. Con los discos pudieron desarrollarse las bases de datos de red y jerárquicas, que permitieron que las estructuras de datos tales como listas y arboles pudieran almacenarse en discos. Los programadores pudieron construir y manipular estas estructuras de datos.
Un artículo histórico de Codd (1970) definió el modelo relacional y formas no procedimentales de consultar los datos en el modelo relacional, y nacieron las bases de datos relacionales. La simplicidad de modelo relacional y la posibilidad de ocultar completamente los detalles de implementación al programador fueron realmente atractivas. Codd obtuvo el prestigioso premio Turing de la ACM (Association of Computing Machinery, asociación de maquinaria informática) por su trabajo.

-Década de 1980

Aunque académicamente interesante, el modelo relacional no se uso inicialmente en la práctica debido a sus inconvenientes por le rendimiento: las bases de datos relacionales no pudieron competir con el rendimiento de las bases de datos de red y jerárquicas existentes. Esta situación cambio con System R, un proyecto innovador de IBM Research que desarrollo técnicas para la construcción de un sistema de bases de datos relacionales eficiente. Los primeros sistemas de bases de datos relacionales, como DB2 de IBM, Oracle, Ingres y Rdb DEC, jugaron un importante papel en el desarrollo de técnicas para el procesamiento eficiente de consultas declarativas. En los principios de la década de 1980 las bases de datos relacionales llegaron a competir con los sistemas de bases de datos jerárquicas y de red incluso en el área de rendimiento. Las bases de datos relacionales fueron tan sencillas de usar que finalmente remplazaron a las bases de datos jerárquicas y de red; los programadores que usaban estas bases de datos estaban forzados a tratar muchos detalles de implementación de bajo nivel y tenían que codificar consultas de forma procedimental. Aun más importante tenían que tener presente el rendimiento durante el diseño de sus programas, lo que implicaba un gran esfuerzo. En cambio en una base de datos relacional, casi todas estas tareas de bajo nivel se realizaban automáticamente por la base de dato, liberando al programador en el nivel lógico. Desde la escala en 1980 el modelo relacional ha conseguido el reinado entre todos los modelos de datos. La década de 1980 también fue testigo de una gran investigación en las bases de datos paralela y distribuida, así como el trabajo inicial en las bases de datos orientadas a objetos.

-Principios de la década de 1990

El lenguaje SQL se diseñó fundamentalmente para las aplicaciones de ayuda a la toma de decisiones, que son intensivas en consultas, mientras que le objetivo principal en las bases de datos en la década de 1980 fue el procedimiento de transacciones, que son intensivas en actualizaciones. La ayuda de la toma de decisiones y las consultas re emergieron como una importante área de aplicación para las bases de datos. Las herramientas para analizar grandes cantidades de datos experimentaron un gran crecimiento de uso.

Muchos vendedores de bases de datos introdujeron productos de bases de datos paralelas en este periodo, así como también comenzaron a ofrecer bases de datos relacionales orientadas a objetos.

-Finales de la década de 1990

El principal acontecimiento fue el crecimiento explosivo de World Wide Web. Las bases de datos se implantaron mucho más extensivamente que nunca antes. Los sistemas de bases de datos tienen ahora soporte de tasas de transacciones muy altas, así como una alta fiabilidad y disponibilidad 24x7 (24 horas x 7 días de la semana). Los sistemas de bases de datos también tuvieron interface web a los datos.

1.2 CONCEPTO DE BASES DE DATOS

Algunas personas conciben a la Base de Datos como un enorme receptáculo en el que un organismo guarda todos los datos procesables que reúne y al cual acuden muy diversos usuarios a su acceso. Este gran almacén puede estar ubicado en una misma localidad o distribuido en varias, todas ellas interconectadas mediante una red de telecomunicaciones. También tienen acceso a la base de datos diversos programas de distinta índole.

Un ejemplo sería: en la Secretaria de Educación Pública (SEP), es un gran organismo, los datos referentes al pago de sus empleados son encontrados en una base de datos diferente a la que guarda los datos escolares de los alumnos de las diferentes escuelas. Además cada una de las Coordinaciones Estatales de este organismo cuanta con una versión de estas dos Bases de Datos.

Por lo tanto tenemos que:

La Base de Datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de manera tal que resulten independientes de los programas que los usan, se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer datos almacenados. Se dice que un sistema comprende una colección de bases de datos cuando estas son independientes desde el punto de vista de su estructura lógica.

Son también un conjunto de información almacenada en memoria auxiliar que permiten el acceso directo y un conjunto de programas que manipulan los datos.

1.3 OBJETIVO DE LAS BASES DE DATOS

1.3.1 Aplicación y Sistemas de las Bases de Datos

Las Bases de Datos son ampliamente usadas. Aquí tenemos algunas aplicaciones más representativas:

Bancos: para la información de clientes, cuentas, préstamos y transacciones bancarias.

Líneas Aéreas: para reservas o información de planificación

Universidades: para información de los estudiantes, matriculas de las asignaturas y cursos.

Transacciones de las Tarjetas de Crédito: para compras y generación mensual de extractos.

Telecomunicaciones: para guardar registros de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.

Finanzas: para almacenar información sobre grandes empresas, ventas y compras de documentos formales financieros, como bolsa y bonos.

Ventas: para información de clientes, productos y compras.

Producción: para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos y pedidos.

Recursos Humanos: parta información sobre el empleado, salarios, impuestos y beneficios, y para la generación de nominas.

Estos son algunos ejemplos de cómo las bases de datos forman parte importante de las empresas. En este siglo las bases de datos han crecido enormemente y esto ayuda a tener un mayor desempeño nivel mundial. En los primeros días muy pocas personas interactuaron directamente con los sistemas de bases de datos, aunque sin darse cuenta interactuaron con bases de datos indirectamente (con informes impresos de las tarjetas de crédito o mediante agentes como cajeros de bancos o agentes de líneas aéreas). Después vinieron los cajeros automáticos y permitieron a los usuarios interactuar con las bases de datos. También la interface telefónica permitió a los usuarios su interacción. Pero la revolución del internet a finales de la década de 1990 aumento el acceso directo a las bases de datos. Las organizaciones convirtieron muchas de sus interfaces telefónicas a las bases de datos en interfaces Web, y pusieron disponibles en línea muchos servicios. Cuando se accede a un sitio web, la información personal puede ser recuperada de una base de datos para seleccionar los anuncios que se deberían mostrar. Los datos sobre los accesos web pueden ser almacenados en una base de datos.

Así aunque las interfaces de datos ocultan detalles del acceso a las bases de datos, y la mayoría de la gente ni siquiera es consciente de que están interactuando con una base de datos, el acceso a las bases de datos forma un aparte esencial de la vida de las personas.

La importancia de los sistemas de bases de datos se pueden juzgar de otra forma: actualmente, los vendedores de sistemas de bases de datos como ORACLE están entre las mayores compañías de software en el mundo, y los sistemas de bases de datos forman parte importante de la línea de productos de compañías más diversificadas como Microsoft e IBM.

Ahora hablemos un poco sobre los sistemas de bases de datos frente a los sistemas de archivos.

Una manera de mantener la información en un computador es almacenarla en archivos del sistema operativo. Este sistema de procesamiento de archivos típico se mantiene mediante un sistema operativo convencional. Los registros permanentes son almacenados en varios archivos y se escriben diferentes programas de aplicación para extraer registros y para

añadir registros a los archivos adecuados. Antes de la llegada de los sistemas de gestión de bases de datos (SGBD), las organizaciones normalmente han almacenado la información usando tales sistemas.

Mantener la información de las organizaciones en un sistema de procesamiento de archivos tiene una serie de inconvenientes importantes:

-Redundancia e inconsistencia de los datos
-Dificultad en el acceso de los datos
-Aislamiento de los datos
-Problemas de integridad
-Problemas de Atomicidad
-Anomalías en el acceso concurrente
-Problemas de Seguridad

Estas dificultades entre otras han motivado el desarrollo de los sistemas de bases de datos.
Mencionaré a continuación cuales son los elementos básicos que deben manejar las bases de datos para poder ser efectivas.

Redundancia Controlada:
La base de datos ha sido definida como una colección no redundante de ítems de datos, pero en realidad en muchas bases de datos se admite cierta redundancia con el fin de disminuir los tiempos de acceso o simplificar los métodos de direccionamiento. Existe la necesidad de armonizar el grado de redundancia con otras características deseables de la base de datos, de modo que es preferible hablar con redundancia controlada.
Con la redundancia controlada, se eliminan eficientemente los inconvenientes de tener tantos datos redundantes que no se pueda tener a tantos de ellos en un mismo grado de actualización, además del eminente costo adicional de almacenamiento y procesamiento de estos datos. Son estos factores los que desacreditan a un sistema de cómputo y lo podrían hacer parecer injustificado, afortunadamente las bases de datos contemplan estos factores sin perjudicar los tiempos de acceso y proceso de los datos.

Independencia de los Datos:
Esta idea implica que los datos y los programas de aplicación que ellos se sirven son mutuamente independientes, de manera que unos puedan ser modificados sin tener en cuenta a los otros. En particular el programador no debe ser afectado por los cambios que sufren los datos en su estructura lógica o física.

Lógico y Físico:
La descripción de los datos y las relaciones que entre ellos existe adopta dos formas: lógica y física. La descripción física de los datos se ocupa de cómo se les registra en el hardware. La descripción lógica a la forma en como se presenta al programador de aplicaciones. Las palabras lógico y físico se usarán para definir otras características de los datos, la primera es para saber como ve los datos el programador y el usuario, y la segunda la forma de cómo se registran en los dispositivos de almacenamiento.

Compartir Datos:
La idea es poseer la capacidad para que varios programas de aplicación compartan en forma independiente una base de datos integra. Es decir, que las representaciones de

datos sean concurrentes y múltiples, que se tengan mecanismo de acceso eficiente para subconjuntos de datos específicos.

Relacionabilidad:
La habilidad para relacionar datos o relacionabilidad se usa para denotar la propiedad de la existencia de relaciones entre diferentes registros lógicos. Un registro representa un concepto del mundo real.

Integridad:
Este se refiere a la coordinación del acceso de datos por programas distintos, a la propagación de valores actualizados a otras copias y valores independientes, y asegura la validez de los datos; incluye también una bitácora en donde se registran todos los accesos y cambios que afecten a cada dato.

Flexibilidad de Acceso:
Se refiere a la capacidad de tener acceso a los datos en forma fácil de diferentes maneras y en base a diferentes llaves de acceso, la capacidad de usar un lenguaje de consulta, y que la base pueda ser accesada por lenguajes convencionales.

Seguridad:
Se refiere a los mecanismos adecuados para asignar derecho de acceso a los datos. Ciertos artículos y combinaciones o selecciones de ellos pueden ser sensibles y requerir de altos niveles da autorización para que se permita su acceso. Un dato debe ser protegido contra actos maliciosos y no autorizados en una base de datos.

1.3.2 Alternativas, Ventajas y Desventajas de las Bases de Datos

Ya hemos hablado que los sistemas de bases de datos no cumplen en si una función mágica y es obsoleto pensar en ellos como la solución a toda clase de problemas, puesto que el enfoque de cada uno de los diferentes paquetes de software disponible en el mercado difiere en cuanto a la orientación específica. Sin embargo pueden señalarse ventajas generales de la organización de bases de datos, así como sus desventajas.

- Los datos podrán accesarse de múltiples maneras
- Se protegerá la inversión intelectual
- Rapidez de desempeño
- Claridad
- Facilidad de acceso
- Flexibilidad
- Rápida atención a problemas no previstos
- Facilidad de cambio
- Precisión y coherencia de datos
- Control de acceso
- Protección contra perdida o daño
- Disponibilidad inmediata para los usuarios
- Independencia física y lógica de datos
- Redundancia controlada
- Fácil recuperación de datos en caso de falla
- Funcionar como ayuda en el diseño y la supervisión

Cuadro 1.1 Objetivos de las bases de datos

VENTAJAS

Del cuadro 1.1 se desprende que:

- No será necesario hacer programas o estructuras lógicas cuando se modifique la base de datos, lo cual si sucedería si se tuvieran los datos en archivos convencionales.
- Se minimizará el costo de almacenamiento de datos, al eliminar la redundancia perjudicial y al existir facilidad de acceso a los mismos datos por diferentes aplicaciones.
- Se reduce el costo de capacitación del personal, debido a que no es necesario para todos los usuarios de los datos, el conocer las complejidades internas, sino solo por el grupo de los administradores de los datos.
- Se evita el exceso de programación, debido a que existen lenguajes de consulta.
- Se evita el acceso no autorizado a los datos. Los datos están protegidos contra fallas o acciones de personas que deseen falsearlos.
- El hardware de almacenamiento de datos y las técnicas de almacenamiento podrán ser modificados, así como el agregado de nuevos ítems de datos y la modificación de la estructura lógica de la base de datos podrán hacerse sin modificar los programas existentes, debido a la independencia lógica y física de los datos.
- Se consigue una normalización de los datos dentro de un organismo, con la finalidad de no generar datos incompatibles, lo que llevará un considerable ahorro de los tiempos de acceso a información entre departamentos con diferentes actividades.

DESVENTAJAS

Quizá la mayor desventaja que se tenga al implementar un sistema de base de datos, sobre todo en un sistema de microcomputadoras, sea el echo de que los sistemas de bases de datos no estén hechos a la medida de los clientes, con lo que la conexión de redes para facilitar el acceso a los datos e vuelve utópica puesto que las redes no son compatibles.

Y posibles inconvenientes que pueden surgir serian:

- la instalación costosa
- el personal especializado
- la implantación larga y difícil
- la falta de rentabilidad a corto plazo
- el desfase entre teoría y practica

1.4 VISION DE LOS DATOS

Un sistema de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que permitan a los usuarios acceder y modificar estos archivos. Un propósito de las base de datos es proporcionar a los usuarios una visión abstracta de lo datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

1.4.1 Abstracción de los Datos

Para que el sistema sea útil necesita recupera los datos eficientemente. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos. Como muchos usuarios de la base de datos no están familiarizados con computadores, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema:

NIVEL FISICO: El nivel mas bajo de la abstracción describe como se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.

En este nivel existen 3 clases de aspectos que deben especificarse:

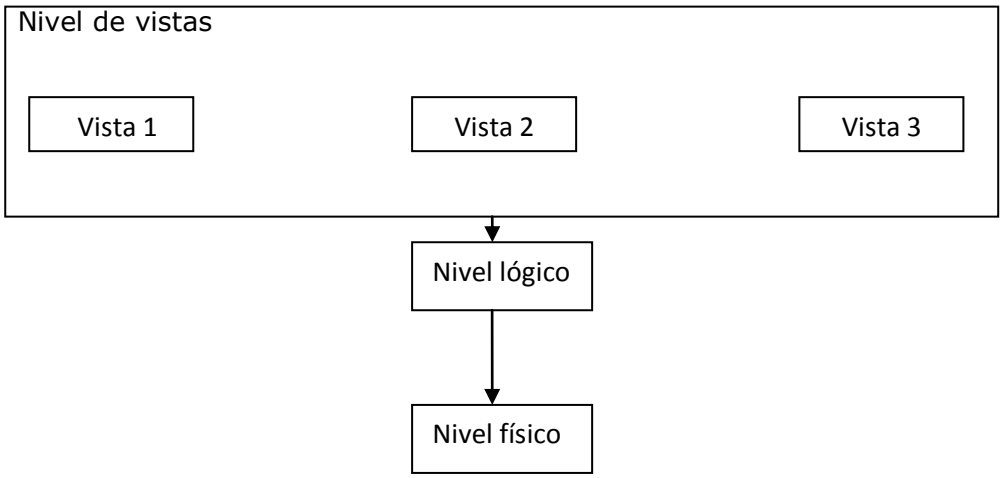
Estrategia de almacenamiento: se incluye la asignación de espacios de almacenamiento para el conjunto de datos. También deberá indicarse la estrategia de emplazamiento de los datos que ha sido utilizada para optimizar tiempos de respuesta y espacio de memoria secundaria; por último deberían de aparecer aspectos como el tratamiento de los desbordamientos.

Caminos de acceso: aquí se ven la especificación de claves, así como la de índices o punteros.

Miscelánea: Aquí habrá que incluir en el esquema interno, otros varios como técnicas de comprensión de datos, de criptografiado, la correspondencia entre esquema interno y esquema conceptual, técnicas de ajuste o afinamiento (tuning), optimización, etc.

NIVEL LOGICO: El siguiente nivel más alto de abstracción describe que datos se almacenan en la base de datos y que relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse por esta complejidad. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.

NIVEL DE VISTAS: El nivel más alto de la abstracción describe solo parte de la base de datos completa. A pesar del uso de estructuras más simples en le nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios del sistema de bases de datos no necesitan toda esta información. En su lugar tales usuarios solo necesitan acceder a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.



El diagrama de la figura 1.2 nos muestra la relación entre los tres niveles de abstracción.

Una analogía con el concepto de tipos de datos en lenguajes de programación puede clarificar la distinción entre los niveles de abstracción. La mayoría de los lenguajes de programación de alto nivel soportan la estructura de tipo registro. Un ejemplo en un lenguaje tipo Pascal, se pueden declarar registros como sigue:

```
Type cliente = record
Nombre-cliente: string;
Id-cliente: string;
Calle-cliente: string;
Ciudad-cliente: string;
End;
```

Este código define un nuevo registro llamado cliente con cuatro campos. Cada campo tiene un nombre y un tipo asociado a el.

En el nivel físico, un registro cliente, cuenta o empleado se puede describir como un bloque de posiciones almacenadas consecutivamente. El compilador del lenguaje esconde este nivel de detalles a los programadores. Análogamente, el sistema de base de datos esconde muchos de los detalles de almacenamiento de nivel inferior a los programadores de bases de datos. Los administradores de bases de datos pueden ser conscientes de ciertos detalles de la organización física de los datos.

En el nivel lógico cada registro de este tipo se describe mediante una definición de tipo como en el ejemplo, y se define la relación entre estos tipos de registros. Los programadores trabajan en este nivel de abstracción y los administradores de bases de datos también.

Finalmente con el nivel de vistas, los usuarios de computadores ven un conjunto de programas de aplicación que esconde los detalles de los tipos de los datos. Análogamente en el nivel de vistas se definen varias vistas de una base de datos y los usuarios de las

mismas ven únicamente y exclusivamente esas vistas. Además de esconder detalles del nivel lógico de las bases de datos, las vistas también proporcionan un mecanismo de seguridad para evitar que los usuarios accedan a ciertas partes de la base de datos.

EJEMPLARES Y ESQUEMAS

Las bases de datos van cambiando a lo largo del tiempo conforme la información se inserta y borra. La colección de información almacenada en la base de datos en un momento particular se denomina un ejemplar de las bases de datos. El diseño completo de la base de datos se llama el esquema de la base de datos. Los esquemas son raramente modificados.

El concepto de esquemas y ejemplares de bases de datos se pueden entender por analogía con un programa escrito en un lenguaje de programación. Un esquema de bases de datos corresponde a la declaración de las variables en un programa. Cada variable tiene un valor particular en un instante de tiempo. Los valores de las variables de los programas en un instante de tiempo corresponden a un ejemplar de un esquema de bases de datos.

Los sistemas de bases de datos tienen varios esquemas divididos de acuerdo a los niveles de abstracción. El esquema físico describe el diseño físico en el nivel físico, mientras el esquema lógico describe el diseño de la base de datos en el nivel lógico. Una base de datos puede tener también varios esquemas en el nivel de vistas, a menudo denominados subesquemas, que describen diferentes vistas de la base de datos.

1.5 MODELO DE LOS DATOS

El modelado de datos es un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes.
A continuación se describirán los modelos de datos.

1.5.1 Modelo Entidad-Relación

Este se puede tomar como representativo de la clase de modelos lógicos basados en objetos, porque éste ha tenido bastante aceptación como modelo de datos apropiado para el diseño de bases de datos y porque se utiliza ampliamente en la práctica.
El modelo de datos Entidad-Relación (E-R), se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y de las relaciones entre estos objetos. Una entidad es un objeto que existe y puede distinguirse de otros. La distinción se logra asociando a cada entidad un conjunto de atributos que describen al objeto.
La estructura lógica general de una base de datos puede expresarse gráficamente por medio de un Diagrama E-R que consta de los siguientes componentes:
Rectángulo, que representan conjuntos de entidades.
Elipse, que representan atributos.
Rombos, que representan relaciones entre conjuntos de entidades.
Líneas, que conectan los atributos a los conjuntos de entidades y los conjuntos de entidades a las relaciones.

Ejemplo:

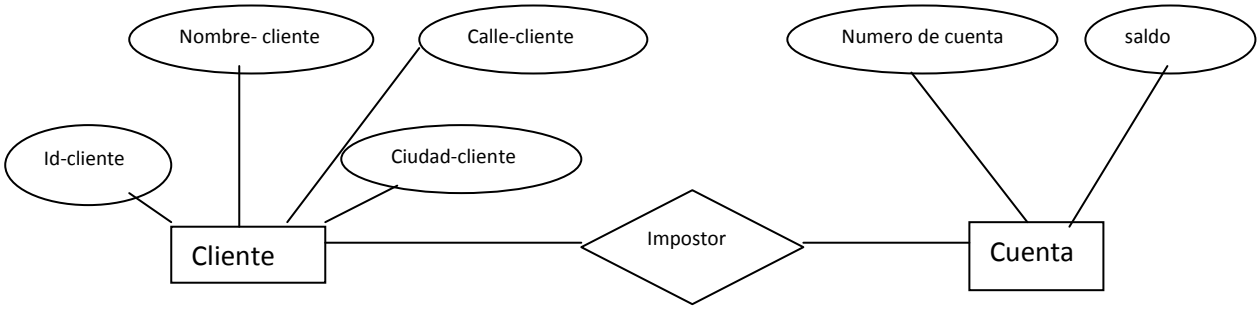


FIGURA 1.3 Diagrama E-R

Este indica que hay dos conjuntos de entidades cliente y cuenta. El diagrama muestra también una relación impostor entre cliente y cuenta.

1.5.2 Modelo Relacional

Modelo en el que la base de datos está constituida por un conjunto de tablas planas o relaciones, en el cual estas expresan por el hecho de que dos relaciones tengan un campo o dominio en común y en el que las relaciones pueden ser 1:N o M:N. cada una de las tablas tiene varias columnas con nombres únicos.

Ejemplo:

Id. Cliente	nombre-cliente	calle-cliente	ciudad-cliente
19.283.746	González	Arenal	La granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peregrinos

Figura 1.4 Modelo Relacional

1.5.3 Otros Modelos

El modelo de datos orientado a objetos es otro modelo de datos que están recibiendo una atención creciente. Este se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos e identidad de objeto.
El modelo de datos relacional orientado a objetos combina las características del modelo de datos orientado a objetos y el modelo de datos relacional. Otros modelos de datos son el modelo de red, el modelo de datos jerárquico y los modelos físicos de datos como el unificador y el de cuadros.

1.6 LENGUAJES DE BASES DE DATOS

1.6.1 lenguaje de definición de datos (LDD)

Un esquema de bases de datos se especifica por medio de una serie de definiciones que se expresan en un lenguaje especial llamado lenguaje de definición de datos LDD. El resultado de la compilación de estas definiciones es una serie de instrucciones que especifica los detalles de implantación de los esquemas de bases de datos que normalmente no pueden ver los usuarios.
El LDD es el lenguaje del sistema generalizado para manejo se bases de datos, usado para definir la estructura lógica de los datos, se tienen 3 lenguajes de LDD:

- el LDD de esquemas para uso del DBA
- El LDD de subesquemas para el uso también del DBA
- El LDD de subesquemas para el uso del usuario

Los LDD de subesquemas para el administrador y es usuario pueden ser muy similares.

1.6.2 Lenguaje de Manejo de Datos (LMD)

Un lenguaje de manejo de datos permite a los usuarios manejar o tener acceso a los datos que estén organizados por medio del modelo apropiado. Existen básicamente 2 tipos de LMD.

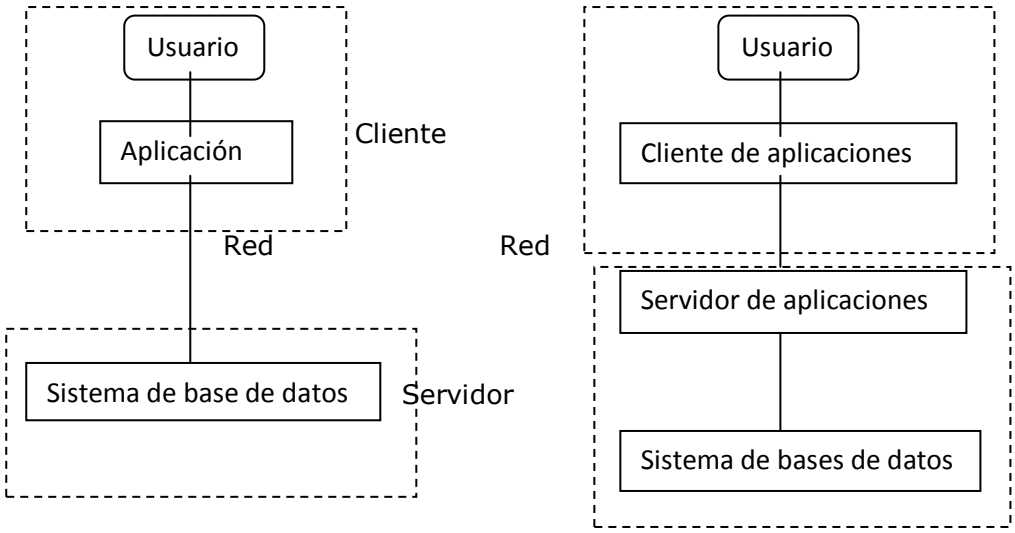
1.- De procedimientos: necesitan que el usuario especifique cuales datos quiere y como debe obtenerse.
2.- Sin procedimientos: requiere que el usuario especifique cuales datos quiere sin especificar como obtenerlos.

El LMD es un conjunto de comandos de un sistema generalizado para manejo de base de datos que se usa para almacenar, recopilar, actualizar, agregar y eliminar datos de una Base de Datos; el LMD incluye a todos los comandos para Entrada/Salida y a otros para navegación en la base. Los comandos del LMD se usan en la división de procedimientos de un programa Cobol. El LMD de un sistema generalizado para manejo de archivos es esencialmente un lenguaje de consultas, diseñado tanto para los requerimentos de la producción de reportes como en instalaciones donde existen solo archivos sencillos y no Bases de Datos.

1.6.3 Arquitectura de un Sistema de Bases de Datos

La mayoría de los usuarios de un sistema de bases de datos no están situados actualmente junto al sistema de bases de datos, sino que se conecta a el a través de una red. Se diferencia entonces desde las maquinas del cliente, en donde trabajan los usuarios remotos de la base y las maquinas servidor, en las que se ejecutan el sistema de bases de datos. Las aplicaciones se dividen usualmente en 2 o 3 capas. En la arquitectura de 2 capas la aplicación se divide en un componente que reside en la maquina cliente, que llama a la funcionalidad del sistema de bases de datos en la maquina servidor mediante instrucciones del lenguaje de consulta. Los estándares de programas de aplicación como ODBC y JDBC se usan para la interacción entre el cliente y servidor.

En cambio en la arquitectura de 3 capas, la máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar el cliente se comunica con un servidor de aplicaciones usualmente mediante una interface de formularios. El servidor de aplicaciones a su vez se comunica con el sistema de bases de datos para acceder a los datos.
La lógica de negocios de la aplicación, que establece las acciones a realizar bajo determinadas condiciones se incorpora en el servidor de aplicaciones, en lugar de ser distribuidas a múltiples clientes. Estas son apropiadas para grandes aplicaciones, y para las que se ejecutan en World Wide Web.



Arquitectura de 2 capas Arquitectura de 3 capas

FIGURA 1.5 Ejemplo de la arquitectura de las bases de datos de 2 y 3 capas.

1.7 USUARIOS Y ADMINISTRADORES DE BASES DE DATOS

Un objetivo principal de un sistema de bases de datos es recuperar información y almacenar nueva información en la base de datos. Las personas que trabajan con una base de datos se pueden catalogar como usuarios de bases de datos o como administradores de bases de datos.

1.7.1 Administrador de Bases de Datos

Una de las razones principales para contar con sistemas de manejo de bases de datos es tener un control centralizado tanto de los datos como de lo programas que tiene acceso a ellos. La persona que tiene el control centralizado del sistema es el Administrador de la Base de Datos (DBA) por sus siglas en inglés.
Es decir el DBA es uno de los individuos que se responsabilizan de la definición de los esquemas, subesquemas, derechos de acceso, niveles de rendimiento, verificaciones de la integridad, etc.; en general, el DBA tiene a su cargo tanto el control y la administración como el uso de la totalidad del Banco de Datos.

1.7.2 Usuarios de Bases de Datos

Típicamente se pueden diferenciar tres diferentes tipos de usuarios de una base de datos.

El diseñador:
Es la persona encargada de hacer el análisis de la información que va a residir en una base de datos, así como de implementar las estructuras lógicas que se usan para el manejo de dicha información, es decir, es aquel que se encarga de "crear" la base de datos.

El administrador:
Aunque se usa el singular, normalmente se trata de un grupo de personas que tiene conocimientos sobre la estructura física y lógica de la base de datos, la cual les permite hacer modificaciones, crear derechos de acceso y en general, controlar el acceso de la información contenida en los archivos de la base de datos. Normalmente estas personas se encargan de asesorar a los usuarios finales respecto a las posibilidades y restricciones del uso de la base de datos en cuestión.

Los usuarios finales:
Como usuarios finales debe conocerse a los programadores de aplicaciones, que se encargan de realizar programas que permiten la utilización de los datos contenidos en la base, y a los usuarios de los programas, típicamente capturistas u operadores sin saber más acerca de ellos.

1.8 GESTION DE TRANSACCIONES

Una transacción es una colección de operaciones que se lleva a cabo como una única función lógica en una aplicación de base de datos. Cada transacción es una unidad de atomicidad y consistencia. Así, se requiere que las transacciones no violen ninguna restricción de consistencia de la base de datos. Es decir, si la base de datos era consistente cuando la transacción comenzó, la base de datos debe ser consistente cuando la transacción termine con éxito.
Es responsabilidad del programador definir adecuadamente las diferentes transacciones, de tal manera que cada una preserve la consistencia en la base de datos. Cada programa en si mismo no trasforma la base de datos de un estado consistente en otro nuevo estado consistente. Así, estos programas no son transacciones.
Asegurar las propiedades de Atomicidad y Durabilidad es responsabilidad del sistema de base de datos, específicamente del componente de gestión de transacciones. En ausencia de fallos, toda transacción completada con éxito y atómica se archiva fácilmente. Sin embargo, debido a diversos tipos de fallos, una transacción puede no siempre completar su ejecución con éxito. Si se asegura la propiedad de atomicidad, una transacción que falle no debe tener efecto en el estado de la base de datos. Así la base de datos se restaura al estado en que estaba antes de la transacción en cuestión comenzara su ejecución. El sistema de base de datos deberá realizar la recuperación de fallos, es decir, detectará los fallos del sistema y restaurar la base de datos al estado que existía antes de que ocurriera el fallo.
Cuando varias transacciones actualizan la base de datos concurrentemente, la consistencia de los datos puede no ser preservada, incluso aunque cada transacción individualmente

sea correcta. Es responsabilidad del gestor de control de concurrencia controlar la interacción entre las transacciones concurrentes para asegurar la consistencia de la base de datos.

1.9 ESTRUCTURA DE UN SISTEMA DE BASES DE DATOS

Un sistema de base de datos se divide en módulos que se encargan de cada una de las responsabilidades del sistema completo. Los componentes funcionales de un sistema de bases de datos se pueden dividir a grandes rasgos en los componentes gestores de almacenamiento y procesador de consultas.
El gestor de almacenamiento es importante porque las bases de datos requieren normalmente una gran cantidad de espacio de almacenamiento. El procesador de consultas es importante porque ayuda al sistema de bases de datos a simplificar y facilitar el acceso a los datos.

1.9.1 Gestor de Almacenamiento

Un gestor de almacenamiento es un módulo de programa que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas del sistema. El gestor de almacenamiento es responsable de la interacción con el gestor de archivos. Los datos en bruto se almacenan en disco usando un sistema de archivos, que está disponible habitualmente en un sistema operativo convencional. El gestor de almacenamiento traduce las diferentes instrucciones LMD a órdenes de un sistema de archivos a bajo nivel. Así el gestor es responsable del almacenamiento, recuperación y actualización de los datos en la base.
Los componentes del gestor de almacenamiento incluyen:

- -gestor de autorización e integridad
- -gestor de transacciones
- -gestor de archivos
- -gestor de memoria intermedia

1.9.2 Procesador de Consultas

Los componentes del procesador de consultas incluyen:

Interprete del LDD, que interpreta las instrucciones del LDD y registra las definiciones en el diccionario de datos.

El compilador de LMD, que traduce las instrucciones del LMD en un lenguaje de consulta en un plan de evaluación que consiste en instrucciones de bajo nivel que entiende el motor de evaluación de consulta.
Este también realiza optimización de consultas, es decir elige un plan de evaluación de menor coste.

Motor de evaluación de consultas, que ejecuta las instrucciones de bajo nivel generadas por el compilador del LMD.

1.10 SISTEMA DE GESTION DE BASE DE DATOS

Los sistemas de gestión de bases de datos (DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz en la base de datos, el usuario y las aplicaciones que utiliza.
Su propósito es el de manejar de manera sencilla, clara y ordenada un conjunto de datos que posteriormente se en información relevante para una organización.

1.10.1 El SGBD como Interfaz entre el Usuario y la Base de Datos

En toda organización se suelen distinguir tres niveles de gestión: operacional, táctico y estratégico, de modo que el sistema de información estará integrado por 3 subsistemas estructurados jerárquicamente y que se corresponden con cada uno de estos tres niveles.
La desconexión entre estos tres subsistemas de información, aumenta el coste global de creación y mantenimiento del sistema de información y produce redundancias e incoherencias; es decir, impide una gestión relacional de los datos.
La base de datos, como deposito único de datos para toda la organización, debe ser capaz de integrar los distintos subsistemas y aplicaciones atendiendo a las necesidades de los usuarios en los tres niveles, siendo el SGBD el que suministra la interfaz entre el conjunto de los datos y los usuarios.

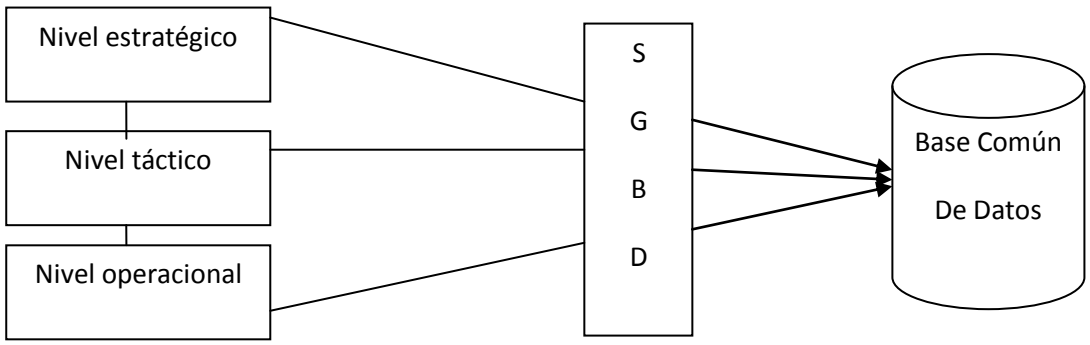


FIGURA 1.6 se muestra el diseño de los 3 niveles de gestión

En el nivel estratégico se elaboran planes y objetivos generales; en el nivel táctico está el control de gestión y los objetivos específicos y en el nivel operacional se manejan las tareas administrativas.

Los distintos tipos de usuarios de una base de datos pueden clasificarse en usuarios informáticos y usuarios finales.

Usuarios Informáticos:

Tienen a su cargo la tarea de creación y mantenimiento de l base de datos, así como la realización de procedimientos y programas que necesiten los usuarios finales. Entre ellos se pueden distinguir:

- Diseñadores: tiene la responsabilidad de identificar los datos que han de estar contenidos en la base de datos, de acuerdo a las necesidades del usuario para satisfacer sus necesidades. Existen 2 tipos:
  Diseñadores lógicos: deben perseguir como objetivo la eficacia de los datos.
  Diseñadores físicos: estos deben optimizar el coste/beneficio.
- Administradores: su misión es la vigilancia y gestión de los datos. El principal recurso en una base de datos son los datos, y el administrador debe velar para que estos no se destruyan ni se contaminen, perdiendo su confidencialidad, disponibilidad e integridad.
  También los administradores tienen a su cargo la gestión de otros recursos distintos de los datos, como pueden ser el SGBD y otras herramientas.
- Analistas y Programadores: tiene a su cargo el análisis y la programación de las tareas que no pueden ser llevadas a cabo por los usuarios finales.

Usuarios Finales:

Son aquellos que tienen acceso a los datos porque los necesitan para llevar a cabo su actividad. Y existen distintos clases de usuarios finales:

- Habituales: suelen hacer consultas o actualizaciones a la base de datos como parte de su trabajo.
- Esporádicos: necesitan la computadora a fin de que les presten una ayuda en su trabajo. Son usuarios a los que hay que suministrar herramientas sencillas y en general potentes ya que en bastantes casos así lo exige su clase de tareas que llevan a cabo.

En resumen, se puede decir que la principal finalidad del SGBD es establecer las adecuadas interfaces entre los diferentes tipos de usuarios y la base de datos.

1.10.2 Concepto y Principales Funciones

Se puede definir el Sistema de Gestión de Base de Datos (SGBD) como un conjunto coordinado de programas, procedimientos, lenguajes, etc. Que suministran a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base, garantizando su seguridad.
El SGBD ha de estar diseñado de forma que las ventajas que se han señalado como propias de las bases de datos sean una realidad. Las operaciones típicas que debe realizar un SGBD pueden resumirse en aquellas que afectan a la totalidad de los datos y las que tienen un lugar sobre registros concretos.

A) Sobre el conjunto de la base
- Creación
- Reestructuración
- Consulta a la totalidad

B) Sobre registros concretos
- Inserción
- Borrado
- Modificación
- Consulta selectiva

} actualización

Las funciones esenciales de un SGBD son las de descripción, manipulación y control.

FUNCION DE DEFINICION O DESCRIPCIÓN

Esta debe permitir al diseñador de la base especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, etc. Así como las características de tipo físico y las vistas de los usuarios.

Esta función realizada por el lenguaje de descripción o definición de datos (LDD) debe suministrar los medios para definir las tres estructuras de datos (externa, lógica, global e interna).

A nivel interno se ha de indicar los espacios reservados para la base, la longitud de los campos o elementos de datos, su modo de representación (binario, decimal, alfanumérico, punto fijo o flotante). Se debe poder definir caminos de acceso, como punteros, índices, etc. La estructura externa y lógica global, la función de descripción ha de proporcionar los instrumentos para la definición de los objetos (tablas, registros, etc.) y su identificación atributos de los mismo, interrelacionados entre ellos, autorizaciones de acceso y restricciones de integridad. Y la estructura lógica está referida a la estructura lógica global. El SGBD, además de suministrar facilidades de descripción, se ocupa de la función de correspondencia o transformación (mapping) de la estructura lógica global a la estructura externa, y a la física.

FUNCION DE MANIPULACION

Una vez descrita la base de datos, es preciso cargar los datos en las estructuras previamente creadas, con los que la base de datos estará ya dispuesta para su utilización. Los usuarios tendrán necesidad, de recupera la información o bien actualizarla.

La consulta a la base de datos puede ser de dos tipos:

Totalidad de los datos o Consulta selectiva En ambos casos se necesitará especificar la estructura lógica externa que se desea recuperar. El SGBD deberá con estos datos acceder a la estructura física de la base de datos, localizar aquellos registros indicados y ponerlos a disposición del usuario. La actualización supondrá tres tipos de operaciones distintas:

Inserción, Borrado y Modificación de Datos

La función de manipulación permite a los usuarios, buscar, añadir o suprimir o modificar los datos de la misma, siempre de acuerdo con las especificaciones y normas dictadas por el administrador. Esta función se lleva a cabo por el lenguaje de manipulación de datos (LMD).

FUNCION DE CONTROL

Esta función debe integrar una serie de instrumentos que faciliten las tareas del administrador. En la mayoría de los SGBD existen funciones de servicio, como cambiar la capacidad de los ficheros, obtener estadísticas de utilización, cargar archivos etc. y principalmente las relacionadas con la seguridad física y de protección frente accesos no autorizados. Todas ellas se consideran comprendidas en esta función de control.

1.10.3 Funcionamiento del SGBD

El funcionamiento esta muy interrelacionado con el de otros componentes, especialmente con el sistema operativo.

Cada SGBD dependiendo de su diseño y de la plataforma a la que se apoye, tiene unas características propias y un modo de funcionamiento específico, por lo que no es posible hacer un análisis pormenorizado de dicho funcionamiento y de su interrelación con el sistema operativo.

Haremos una comparación de la manera en los programas de aplicación interaccionan con los datos de un sistema de ficheros y una base de datos.

El primer caso, los datos se organizan por ficheros cada uno se diseña con objeto de atender una determinada aplicación periódica. Las aplicaciones eventuales suelen tener necesidad de crear nuevos ficheros, temporales o permanentes.

Cuando se trata del enfoque hacia la base de datos, se organizan en un conjunto estructurado sin redundancias perjudiciales, y que son utilizadas por todas las aplicaciones, ya sean eventuales o periódicas.

Cuando se trata de una base de datos el programa escrito en un lenguaje anfitrión como llamadas en el LMD, se dirige al SGBD, el cual accede a la base de datos a través del sistema operativo.

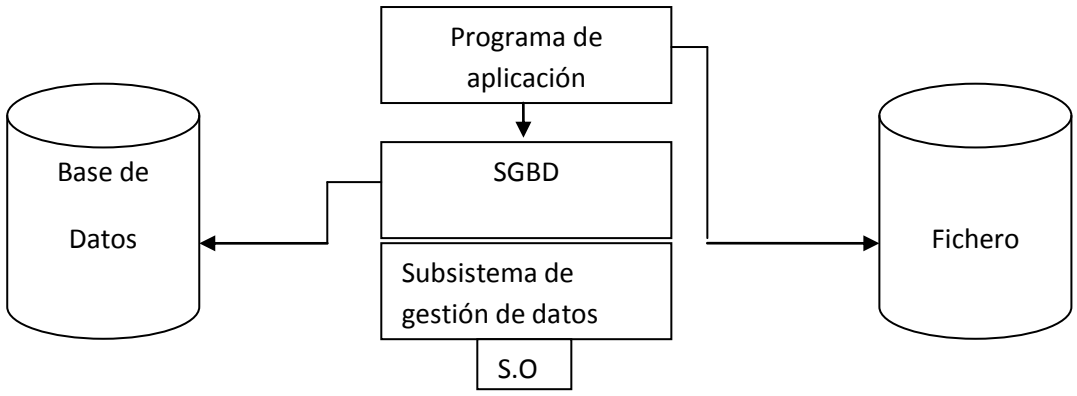


Figura 1.7 Aquí se muestra la interacción en un entorno concurrente en el SGBD, el sistema operativo y los programas de aplicación.

1.10.4 Estructura General del SGBD

Mostraremos la estructura de un SGBD donde además del núcleo del sistema, existe un conjunto de facilidades y herramientas que pueden ser proporcionadas, por el suministrador del SGBD o bien por los vendedores independientes.

En la figura 1.8 se muestra el núcleo del SGBD, que está en mayor o menor medida soportado por el sistema operativo; sobre este se sitúa el diccionario llamado también meta base. El conjunto de herramientas y facilidades que aparecen en la figura facilitan el acceso a los datos, sea directamente o mediante aplicaciones desarrolladas por los informáticos con la ayuda de pre compilador, generador de aplicaciones, etc.

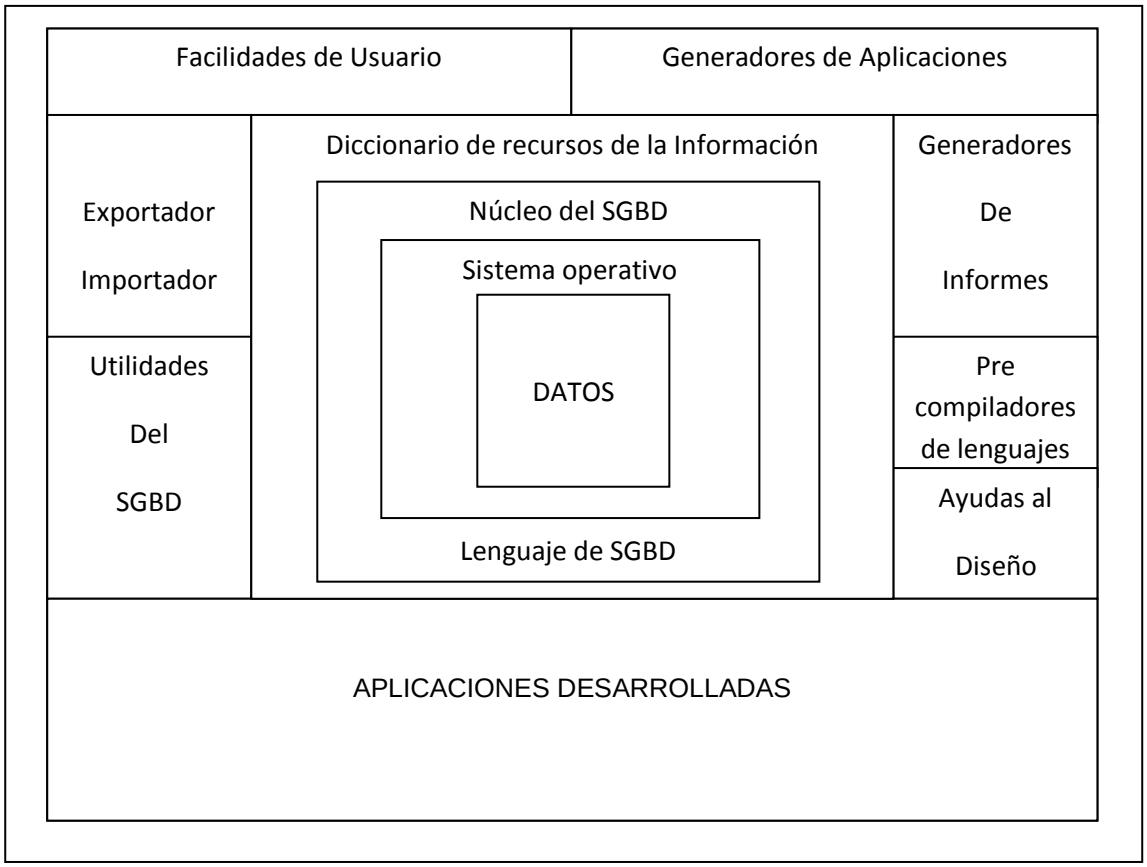


Figura 1.8 Núcleo de un SGBD y conjunto de herramientas

CAPITULO 2. TERMINOS GENERALES DE SEGURIDAD INFORMATICA

2.1 ANTECEDENTES

Desde la aparición del hombre, la información a tenido gran importancia en cualquier actividad. Sin duda, cuando esta actividad tiene propósitos muy especiales, como el militar o personal, la información crece en importancia y requiere mayor atención en su resguardo. La información, como elemento indispensable en la comunicación, se puede clasificar en varios niveles dependiendo su valor, por ejemplo existe información confidencial en las actividades militares, en las actividades comerciales importantes, en las transacciones financieras, etc. Una forma de poder dar seguridad a toda esta información es implementar y usar diversos métodos de control de acceso a ella.

Los incidentes de seguridad en cómputo a nivel mundial aparecieron desde la invención de los primero sistemas de cómputo. En aquellos años era muy difícil que quien quisiera penetrar esos sistemas tuviera que ir personalmente y realizarlos desde la consola del sistema, por lo que se reducía el índice de probabilidad de que dicho ataque pudiera venir de personas de afuera o no perteneciente a las empresas o ambientes académicos.

Los sistemas de cómputo son herramientas muy poderosas que actualmente han adquirido una importancia insospechada hace tan solo unas décadas. Se ha llegado a depender de las computadoras como nunca antes se hizo de ningún otro dispositivo electrónico, pues gran parte de los datos que nosotros, a las entidades de nuestra sociedad manejamos han sido tratados, sea durante su almacenamiento, proceso o transición mediante las llamadas tecnológicas de la información entre las que se ocupa un lugar focal en la informática.

Por consecuencia, la seguridad de las tecnologías de información se convierte en un tema de crucial importancia para el continuo y espectacular progreso de la sociedad e incluso para la propia supervivencia. Motivo por el cual se hace de vital necesidad establecer políticas y lineamientos de acceso a estos dispositivos que manipulan y almacenan datos de relevada importancia para la mayoría de las personas actualmente; pues el problema reside en nosotros mismos.

Paradójicamente nos hemos convertido en el enemigo más común para los sistemas de cómputo actuales, por razones de naturaleza humana: la envidia, la falta de ética, los celos profesionales, la venganza, la avaricia, la inconformidad, etc. Son las principales causas de los incidentes de seguridad de cómputo que suceden hoy en día.

Los avances tecnológicos que día a día se van logrando no se han mantenido distantes del mundo del cómputo, por el contrario han ido a la par y constantemente se van logrando avances significativos que hacen posible tener computadoras mas sofisticadas, equipos de cómputo mejores, redes de computo más veloces, etc. Hace 20 años cuando las computadoras aun no estaban conectadas una con otra y el internet era solo un proyecto de unos cuantos, era prácticamente difícil creer en incidentes de seguridad. Llega 1983 y con él, el protocolo de comunicación TCP/IP que trajo consigo una posible comunicación entre sistemas de cómputo, las distancias se acortaron, los sistemas de cómputo cambiaron den ser simples redes de área local (LAN) se extendieron a uso metropolitano (MAN) e incluso de alcance mundial (WAN); surgió la tendencia cliente- servidor, los sistemas de cómputo se unieron y comenzó la integración del mundo gracias a la tecnología y a las redes de computadoras. En ese entonces no se pensaba en individuos que pudieran acceder a sistemas de cómputo remotos y mucho menos que pudieran pensar que pudieran causar daño desde distancias lejanas. Con el paso del tiempo, cada vez fue más notable lo inseguro que eran los sistemas y que tanto hardware como software contenía fallas de elaboración y de programación.

Los primeros incidentes de seguridad llegaron y con ellos los múltiples problemas; pero no había legislación, no existía ningún organismo que fuera el responsable de denunciarlos ni que pudiese hacer algo al respecto. En aquellos años a finales de los 80s la mayoría de los crackers utilizaba técnicas tan triviales como el adivinar el login y de saber nombre del usuario, tratar de perpetrar el sistema con contraseñas fáciles de adivinar, siendo esta una tarea trivial y que hoy en día aun la practican.

Posteriormente surgieron incidentes de seguridad a nivel mundial; por ejemplo, el 28 de diciembre de 1998 un grupo de crackers norteamericanos la *Legion of the Underground*, declaró la ciberguerra contra Irak y China, amparándose en que dichos países no respetan los derechos y libertades fundamentales, llamaron a la destrucción masiva de todas las redes informáticas de estos países. Su primer victima fue el servidor oficial Iraquí, que sucumbió el 7 de enero.

Sin embargo el resto de la comunidad hackers se opone frontalmente a este tipo de medidas. En el manifiesto que estos otros grupos publicaron, declararon oponerse totalmente a cualquier intento de usar el poder del hacking para amenazar o destruir las infraestructuras de comunicación de cualquier país, por cuanto las redes de comunicaciones son el sistema nervioso de nuestro planeta. También a raíz del bombardeo de la embajada china en Belgrado (mayo 1999), los internautas chinos inundaron la red con consignas en contra de los Estado Unidos, entraron en la Web de la embajada estadounidense y colapsaron las charlas en directo condenando las acciones de la alianza.

Este tipo de situaciones dieron pauta a la creación del máximo organismo de Seguridad en los EU. Y en 1988 se creo el CERT *(Computer Emergency Response Team)* a la cual se unieron el FIRST (*Forum of Incident and Security Teams*) y el CIAC (*Computer Incident Advisory Capabilities)*.

También se dio la creación de sucursales de equipos de respuesta a los incidentes de Seguridad en casi al mayoría de los países que no estaban aislados a los cambios tecnológicos constantes como lo fue en 1995 la creación en nuestro país de la sucursal de CERT, llamado MX-CERT. Desgraciadamente casi ninguna institución tiene políticas ni procedimientos, nadie sabe que es lo

permitido ni que es lo prohibido. Es muy alarmante encontrar que salvo las grandes corporaciones, la normatividad relativa a la tecnología de información es prácticamente inexistente, no hay lineamientos establecidos para administrar recursos como el correo electrónico, los mecanismo de seguridad, los niveles de servicio en redes y la atención de problemas.

La cultura de la seguridad informática es entonces un concepto que debe cubrir todos los niveles jerárquicos, así como todas las funciones que la conforman.

INTERNET

Es hoy en día una palabra reconocida por millones de personas en el mundo y su uso se ha convertido en un componente medular de la vida. La gente se informa y se comunica a través de la internet; conforme aumenta el uso de la empresas y corporaciones, las oportunidades de negocios que se desarrollan son cada vez mayores y más interesantes; se anuncian y venden sus productos por este medio. Sin embargo el crecimiento de las amenazas es igualmente acelerado, en este mundo cada vez más globalizado, con la internet se mueven millones de datos financieros.

El Web está formado por varios componentes:

- Los servidores, en los que los sitios conectados a internet pueden exportar datos al mundo.
- Los clientes con los que los usuarios pueden navegar
- Los proxies que se utilizan para facilitar la comunicación con internet, como pudieran ser sitios atrás de firewall.

Se comunican decisiones y se efectúan transacciones a ritmos cada vez más acelerados. Un problema importante con los sistemas inalámbricos es que hoy un gran número de usuarios comparten un canal común, lo que genera conflictos en la privacidad y seguridad de la información.

La red de internet conecta a millones de personas en el mundo, abriendo nuevos canales de comunicación, nuevas oportunidades, nuevos negocios, pero abriendo también la disponibilidad de invadir las redes corporativas. Para enfrentar este problema es vital contar con herramientas que den seguridad a toda prueba.

Las conexiones internet dedicadas, requieren incorporar sistemas que:

- Den seguridad a la red interna de la empresa
- Autentifiquen el ingreso de usuarios remotos
- Optimicen el uso de internet por parte de los usuarios internos
- Generen canales encriptados a través de la red para permitir el intercambio de información en forma confiable entre distintos grupos.

2.2 CONCEPTOS BASICOS

INFORMACION Y SISTEMAS INFORMATICOS

Entendemos por información el conjunto de datos que sirven para tomar una decisión. La información también es necesaria para el estudio de las desviaciones y de los efectos de las acciones correctoras, es un componente vital para el control.

Se puede ver el sistema informático como el conjunto de los recursos técnicos, financieros y humanos, cuyo objetivo consiste en el almacenamiento, procesamiento y transmisión de la información de una organización.

Aspectos claves en la Seguridad de los Sistemas de Información (SSI)

En primer término con la expansión del uso de computadores personales se ha magnificado el problema de la SSI, debido sobretodo a la carencia de controles de seguridad básicos en este tipo de sistema. En segundo lugar, la evolución hacia entornos con acceso global y múltiple, con un aumento en la conectividad entre organizaciones distintas, plantea retos importantes a la gestión de la seguridad.

Los riesgos fundamentales asociados con la incorrecta protección de la información son:

Revelación a personas no autorizadas

Inexactitud de los datos

Inaccesibilidad de la información cuando se necesita

Estos aspectos se relaciona con las tres características que debe cubrir un sistema de información seguro: confidencialidad, integridad y disponibilidad. Así pues preservar estas tres características de la información constituyen el objetivo de la seguridad.

Seguridad Informática

No existe una definición estricta de lo que es seguridad informática, puesto que ésta abarca múltiples y diversas áreas relacionadas con los sistemas de información. Áreas que van desde la protección física de la maquina como componentes hardware de su entorno; hasta la protección de la información que contiene o de las redes que lo comunican con el exterior.

Tampoco es único el objetivo de la seguridad. Son muy diversos tipos de amenaza contra los que se deben de proteger, desde amenazas físicas, como los cortes eléctricos, hasta errores no intencionados de los usuarios, pasando por los virus informáticos o el robo, destrucción o modificación de la información.

2.3 VULNERABILIDAD, AMENAZAS Y CONTRAMEDIDAS

VULNERABILIDAD

Punto o aspecto del sistema que es susceptible de ser atacado o de dañar la seguridad del mismo. Representar las debilidades o aspectos fiables o atacables en el sistema informático.

AMENAZAS

Posible peligro del sistema. Puede ser una persona (cracker), un programa (virus, caballo…) o un suceso natural o de otra índole (fuego, inundación…). Representa los posibles atacantes o factores que aprovechan las debilidades del sistema.

CONTRAMEDIDA

Estas son las técnicas de protección del sistema contra las amenazas.

La seguridad informática se encarga de la identificación de la vulnerabilidad del sistema y del establecimiento de contramedidas que eviten que las distintas amenazas posibles exploten dichas vulnerabilidades. Una máxima de la seguridad informática es que "No existe ningún sistema completamente seguro". La seguridad nunca es absoluta.

2.3.1 Tipos de Vulnerabilidad

Algunos tipos son los siguientes:

Vulnerabilidad Física: se encuentra en el entorno del sistema, se relaciona con la posibilidad de entrar o acceder fácilmente al sistema para robar, modificar o destruir el mismo.

Vulnerabilidad Natural: se refiere al grado de que los sistemas son afectados por desastres naturales.

Vulnerabilidad del hardware y del software: En el hardware cierto tipo de dispositivos pueden ser más vulnerables que otros. Así en ciertos sistemas es necesario contar con algún tipo de herramienta o tarjetas para poder acceder al mismo.

Ciertos fallos o debilidades en el software del sistema, hacen más fácil acceder al mismo y lo hacen menos fiable.

Vulnerabilidad a los medios o dispositivos: se refiere a la posibilidad de dañar o dañar los discos, cintas, listados de impresoras o cualquier otro medio que contenga la información.

Vulnerabilidad por Emanación: todos los dispositivos eléctricos o electrónicos emiten radiaciones electromagnéticas. Existen dispositivos encargados de interceptar estas emanaciones y descifrar o reconstruir la información almacenada o transmitida.

Vulnerabilidad de las comunicaciones: la conexión de las computadoras a redes supone sin duda un enorme incremento de la vulnerabilidad del sistema. Se puede penetrar al sistema a través de la red e interceptar información que es transmitida desde o hacia el sistema.

Vulnerabilidad humana: la gente que administra y utiliza el sistema, representa la mayor vulnerabilidad del mismo. Toda la seguridad del sistema recae en el administrador y el debe proteger el sistema al máximo contra posibles ataques o incidentes intencionales.

2.3.2 Tipos de Amenazas

Las amenazas al sistema informático, pueden también clasificarse desde varios puntos de vista. En una primera clasificación, según el efecto causado en el sistema, las amenazas pueden englobarse en cuatro grandes tipos.

Intercepción: es cuando una persona, programa o proceso logra una entrada no autorizada al sistema.

Modificación: se trata de acceder al sistema no solo sin autorización sino también modificando los datos del mismo.

Interrupción: interrumpir el sistema de algún modo posible.

Generación: se refiere a la posibilidad de añadir información o programas al sistema.

Desde el punto de vista del origen de las amenazas se pueden clasificar en:

Naturales: las que ponen en peligro los componentes físicos del sistema, como son los desastres naturales y por otro las condiciones ambientales.

Involuntarias: están relacionadas con el uso descuidado del equipo por falta de: borrar por descuido total la información, dejar sin protección o sin respaldo los archivos básicos del sistema, dejar el password o contraseña a la vista de todos.

Intencionadas: son las que se pretende entrar al sistema para borrar, modificar o robar la información, para bloquear o por simple diversión.

2.3.3 Tipos de Medidas de Seguridad o Contramedidas

Diseñar sistemas mediante criterios de seguridad es más complejo, pues las amenazas son en la mayoría de las situaciones poco cuantificables y muy variadas. En muchos casos las medidas de seguridad llevan un costo aparejado que obliga a subordinar algunas de las ventajas del sistema: por ejemplo, la velocidad de las transacciones. Con relación a esto, también se hace obvio que a mayores y más restrictivas medidas de seguridad, menos amigable es el sistema, se hace menos cómodo para el usuario ya que limita su actuación y establece unas reglas más estrictas que a veces dificultan su manejo.

Las medidas que se pueden establecer en un sistema informático, son de 4 tipos fundamentales:

Medidas físicas:

Se trata fundamentalmente de establecer un perímetro de seguridad en el sistema. Se aplican mecanismos para impedir el acceso directo o físico no autorizado al sistema. También se le protege de desastres naturales o condiciones medioambientales adversas.

Los tipos de controles que se establecen incluyen:

-control de las condiciones medioambientales

-prevención de catástrofes

-vigilancia

-sistemas de contingencia

-sistemas de recuperación

-control de entrada y salida de material.

Medidas lógicas:

Se refieren más a la protección de la información almacenada. Incluye las medidas de acceso a la información, y a su uso correcto; así como a la distribución de las responsabilidades entre los usuarios. Dentro de estas se incluyen también las que se refiere a las personas y que se pueden denominar medidas humanas. Se trata de definir las funciones, relaciones y responsabilidades de los distintos usuarios potenciales del sistema; para ello se debe tomar en cuanta, el tipo de usuarios de que se trata, porque a cada uno se les aplica una política de control de acceso diferente y se le imputará distinto grado de responsabilidad sobre el sistema.

Se diferencian 4 tipos de usuarios:

-El administrador del sistema

-Los usuarios del sistema

-Las personas relacionadas con el sistema pero sin necesidad de usarlo

-Las personas ajenas al sistema

Entre los tipos de controles lógicos que es posible incluir en una política de seguridad, se pueden destacar los siguientes:

1.- El establecimiento de una política de control de acceso. Incluyendo un sistema de identificación y autentificación de usuarios autorizados y un sistema de control de acceso a la información.

2.- Definición de un apolítica de instalación y copia de software.

3.- Uso de la criptografía para proteger los datos y las comunicaciones.

4.- Uso de contrafuegos para proteger una red local de internet.

5.- Definición de una política de copias de seguridad.

6.- Definición de una política de monitoreo (loggin) y auditoria (auditing) del sistema.

Medidas Administrativas:

Son aquellas que deben ser tomadas por las personas encargadas de definir las políticas de seguridad para ponerlas en práctica, hacerlas viables y vigilar su correcto funcionamiento. Algunas de las medidas administrativas fundamentales a tomar son las siguientes:

º Documentación: y publicación de políticas de seguridad y de las medidas tomadas para ejercerlas.

º Debe quedar claro quien fija la política de seguridad y quien le pone en práctica.

º Establecimiento de un plan de formación del personal. Los usuarios deben tener los conocimientos técnicos necesarios para usar la parte del sistema que les corresponda, y de esta manera evitar toda una serie de fallos involuntarios que puedan provocar graves problemas de seguridad.

º Los usuarios deben ser consientes de los problemas de seguridad de información a la que tienen acceso.

º Deben conocer las políticas y las medidas de seguridad tomadas para colaborar poniéndose en práctica.

º Los usuarios deben conocer sus responsabilidades respecto al uso del sistema informático, y deben ser consientes de las consecuencias de un mal uso de este.

Medidas Legales:

Se refiere a la aplicación de las medidas legales para disuadir al posible ataque o para aplicarle algún tipo de castigo. Este tipo de medidas trascienden el ámbito de la empresa y normalmente son fijadas por instituciones gubernamentales e incluso instituciones internacionales.

2.3.4 Políticas de Seguridad

La política de seguridad es una declaración de intenciones de alto nivel, que cubre la seguridad de los sistemas de información y que proporciona las bases para definir y delimitar responsabilidades para las diversas actuaciones técnicas y organizativas que se requerirán.

La política se refleja en una serie de normas, reglamentos y protocolo a seguir, donde se definen las distintas medidas a tomar para proteger la seguridad del sistema, las funciones o responsabilidades de los distintos componentes de la organización y los mecanismos para controlar su correcto funcionamiento.

Son los directivos, junto con los expertos en tecnologías de la información, quienes pueden definir los requisitos de seguridad, identificando y priorizando la importancia de los distintos elementos de la actividad realizada; con lo que los procesos más importantes recibirán más protección.

Para lograr el éxito en la realización e implantación de las políticas, la seguridad debe considerarse como parte de la operativa habitual y no como un extra añadido.

Algunas reglas básicas para establecer una política de seguridad, se mencionan a continuación.

- Todas las políticas de seguridad deben ser holísticas, es decir, deben de cubrir todos los aspectos relacionados con el sistema:

  1.- Debe proteger el sistema en todos los niveles; físico, humano y lógico.

  2.- Debe tener en cuanta no solo los componentes del sistema, tales como el hardware, software, entorno físico y usuarios, sino también la interacción entre los mismos.

  3.- Deben tener en cuenta el entorno del sistema, esto es, el tipo de compañía o entidad de que se trate.

- La política de seguridad debe adecuarse a las necesidades y recursos con que se cuente, al valor que se le da a los recursos y a la información, al uso que se hace del sistema en todos los departamentos.

  1.- Deben evaluarse los riesgos, el valor del sistema protegido y el costo de atacarlos. Las medidas de seguridad tomadas deben ser proporcionales a estos valores.

- Todas las políticas de seguridad deben basarse fundamentalmente en el sentido común, para esto es necesario contar con:

1.- Conocimientos del sistema a proteger y de su entorno.

2.- Conocimientos y experiencia en la evaluación de riesgos y el establecimiento de las medidas de seguridad.

3.- Conocimientos de la naturaleza humana, de los usuarios y de sus posibles motivaciones.

Este es un resumen de los pasos básicos:

Análisis de Riesgo:

1. Determinar los recursos a proteger y su valor
2. Analizar las vulnerabilidades y amenazas del sistema, su probabilidad y costo.

Gestión de Riesgos:

3.- Definir las medidas a establecer para proteger el sistema.

4.- Vigilar el cumplimiento de la política, revisarla y mejorarla cada vez que se detecte un problema.

Estas medidas deben establecerse para todos los niveles: físico, lógico y humano. Además deben definir una estrategia en caso de fallos.

Existen también 3 tipos de políticas:

Política administrativa:

Se encarga de los procedimientos administrativos relacionados con la seguridad, no de los aspectos técnicos ni ejecución de esta. Estos son los puntos que debe de tratar la política administrativa.

Análisis y gestión de riesgos
Políticas de Actuación en caso de desastre
Monitoreo y audición de los sistemas y de los empleados
Capacitación y formación de los usuarios
Política de copia de seguridad

Políticas de control de accesos:

Estable bajo que condiciones cada sujeto puede acceder a cada objeto. Se entenderá en este caso por sujeto a cualquier usuario, programa, computadora remota u otro dispositivo que pueda tener acceso al sistema. Se entenderá por acceso cualquier acción aplicada sobre los objetos, tal como leer, escribir, modificar o ejecutar. Finalmente se entenderá por objeto cualquier archivo, directorio, proceso en memoria, dispositivo, etc. del sistema. Existen distintos criterios para las políticas de control de acceso:

-Política de menor privilegio o de necesidad del saber
-Política de compartición
-Granularidad. Es el tamaño mínimo de los objetos accesibles
-Políticas cerradas
-Políticas abiertas

Políticas de control de flujo

Estas tratan sobre la difusión de la información una vez accedida, estableciendo cuales son los canales legítimos para su difusión. Al hablar de canales, no se hace referencia siempre a canales físicos de transmisión de información, sino a sistemas de transferencia, a las distintas formas en las que la información puede fluir de un sujeto origen a un sujeto destino.

Una de las primeras elecciones de toda política de control de flujo es la prioridad que se le da a los tres aspectos de la seguridad: confidencialidad, integridad y disponibilidad.

2.4 CONFIDENCIALIDAD

En un SGBD, al igual que sucede con el sistema operativo, existen diverso elementos básicos que ayudan a controlar el acceso de los datos. En primer lugar, el sistema debe identificar y autenticar al usuario, utilizando para ello algunos mecanismos, MORANT et al. (1997):

- Código y contraseña
- Identificación por hardware
- Características bioantropométricas (huellas dactilares, voz, retina de ojo, palma de la mano, etc...)
- Conocimientos, aptitudes y hábitos del usuario
- Información predefinida (aficiones, datos culturales, personales).

Como sabemos, la forma más usual es la primera en la que los usuarios dan su identificación, el SGBD le pide la contraseña y le concede el acceso al sistema, si ambos son válidos.

El administrador debe definir que tipo de privilegios tendrá el usuario sobre la base de datos. Estos privilegios incluyen, entre otros:

-Utilizar una base de datos

-Consultar ciertos datos.

-Actualizar datos

-Crear o actualizar objetos

-Ejecutar procedimientos almacenados

-Referenciar objetos

-Indexar objetos

-Crear identificadores

-Conceder privilegios

Para facilitar la administración de la confidencialidad, los SGBD suelen incorporar el concepto de perfil, rol o grupo de usuarios, que agrupa una serie de privilegios de forma que el usuario asignado a un grupo, herede todos los privilegios del grupo.

Con esta información, el mecanismo de control de acceso se encarga de denegar o conceder el acceso a los usuarios, ayudando a mantener la integridad de los datos, en caso de tratarse de operaciones de actualización.

En un SGBD hay que tener en cuenta que pueden existir diferentes tipos de autorización. Como:

Autorización explicita, que utilizan normalmente los sistemas tradicionales, que consiste en almacenar que sujetos pueden acceder a ciertos objetos con determinados privilegios; para ello se suele utilizar una matriz de control de accesos.

Autorización implícita: consiste en que una autorización definida sobre un objeto puede definirse a partir de otras.

Autorización fuerte, en caso de que las autorizaciones deducidas a partir de la misma no puedan ser invalidadas.

Autorización débil, en este caso se permiten excepciones sobre las autorizaciones implícitas.

Autorización positiva, su presencia indica la existencia de autorización

Autorización negativa, es la denegación explicita de una autorización

El tipo de autorización que se adopte dependerá de otras cosas como:

La política de control elegida, pudiendo el sistema operar como un sistema abierto, o como un sistema cerrado.

Los sistemas de bases de datos actuales suelen proporcionar lo que se denomina control de acceso discrecional en el que, como hemos visto, son los usuarios los encargados de establecer el control, principalmente, a través de privilegios. Este tipo de seguridad es suficiente para un gran número de sistemas, pero algunas aplicaciones y determinados organismos requieren además un nivel superior de seguridad que se denomina control de acceso obligatorio, que ofrecen los denominados SGBD multinivel y que se han incorporado ya en las últimas versiones de productos como DB2.

Un sistema de gestión de base de datos multinivel soporta datos con diferentes niveles o clases de confidencialidad y usuarios con diferentes clases de autoridad. Una clase de confidencialidad consta de 2 componentes: uno jerárquico (Alto secreto, secreto, confidencial, no clasificado) junto a un conjunto de categorías no jerárquicas. La diferencia con respecto a la seguridad discrecional radica en que los datos tiene un nivel de seguridad por si mismos, con independencia de los que se atribuyan a los usuarios. Los sistemas de gestión de bases de datos multinivel utilizan las clases de seguridad para implantar el control de acceso obligatorio.

El control de acceso obligatorio se basa en 2 reglas, definidas originalmente en el modelo Bell-LaPadula:

1.- Regla de lectura –no-ascendente (no-read-up), también denominada propiedad de seguridad simple, se encarga de proteger los datos contra accesos no autorizados.

2.- Regla de escritura-no-descendente (no-write-Down), que algunos denominan propiedad "*" (estrella); se ocupa de la protección de datos contra su contaminación.

En definitiva la responsabilidad de un SGBD multinivel es asegurar que cada usuario obtiene acceso, directa o indirectamente, solo aquellos datos para los que este autorizado.

De todos modos no basta con las reglas de seguridad mencionadas, sino que a demás hay que proteger los datos contra su revelación a través de canales encubiertos.

Los SGBD siguen la arquitectura a tres niveles ANSI/SPARC, soportan los esquemas externos, que también permiten controlar el acceso a los datos por parte de los usuarios.

Otra técnica que también se puede utilizar es la criptografía, que permite transformar en contenido de la base, haciéndola ininteligible a cualquier usuario que acceda a la misma sin la correspondiente clave de descifrado.

2.5 DISPONIBILIDAD

Los sistemas de bases de datos deben asegurar la disponibilidad de los datos aquellos usuarios que tiene derecho a ello, por lo que proporcionan mecanismo que permiten recuperar la base de datos contra fallos lógicos o físicos, que destruyan los datos en todo o en partes.

Estos fallos van desde catástrofes como un incendio o un terremoto, a sabotajes, fallos del sistema operativo, fallos del disco duro u otras caídas del sistema sea cual sea la causa que los haya provocado. Nos preocuparemos solo de los instrumentos que proporcionan el propio SGBD para evitar o remediar estos fallos aunque hay que ser consientes de que para obtener un sistema robusto podría ser conveniente, bajo determinadas circunstancias, utilizar facilidades ajenas al SGBD.

El principio básico en el que se apoya la recuperación de la base de datos ante cualquier fallo es la redundancia física para el usuario que accede a la base de datos.

En lo que afecta al SGBD existen 2 tipos importantes de fallos:

- Los que provocan la perdida de memoria volátil, usualmente por interrupción de suministro eléctrico o por mal funcionamiento del hardware.
- Los que provocan la perdida de contenido de memoria secundario, por ejemplo cuando se patinan las cabezas de un disco.

2.5.1 Concepto de Transacción

Lo importante ante cualquier tipo de fallo es asegurar que después de una actualización, la base de datos queda en un estado inconsistente. Para conseguir esto se crean unidades de ejecución denominadas transacciones, que pueden definirse como secuencias de operaciones que han de ejecutarse de forma atómica, es decir, o bien se realizan todas las operaciones que comprenden la transacción globalmente o bien no se realiza ninguna.

El ejemplo clásico de una transacción es la de una operación bancaria de transferencia de dinero entre dos cuentas corrientes, en la cual se sustrae dinero de una cuenta o se añade a otra, o bien no se lleva a cabo ninguna operación, pero lo que hay que impedir es que por un fallo del sistema se restase el dinero de una cuenta sin llegar a sumarlo a otra.

Por definición, la base de datos se encuentra en un estado consistente antes de que se empiece a ejecutar una transacción y también lo deberá estar cuando la transacción termine de ejecutarse.

Las propiedades principales de las transacciones son las siguientes:

Atomicidad: se ejecutan todas las sentencias o ninguna

Preservación de la consistencia: la ejecución de una transacción debe dejar a la base de datos en un estado consistente.

Aislamiento: es por que la transacción no muestra los cambios que produce hasta que finaliza.

Persistencia: cuando finaliza la transacción, sus efectos perduran en la base de datos.

Una transacción puede terminar de dos maneras distintas:

Con éxito, en cuyo caso las actualizaciones de que consta la transacción se graban (commit), esto es, se hacen permanentes.

Con fracaso, en cuyo caso debe ser restaurado el estado inicial en el que se encontraba la base antes de que empezara a ejecutarse la transacción. Las actualizaciones de que constan las transacciones deberán, por tanto, deshacerse (rollback).

Los SGBD suelen seccionar las transacciones de forma implícita, ofreciendo además al usuario para la gestión explicita de transacciones.

Existen componentes del SGBD que se encargan de la gestión y recuperación de las transacciones:

Gestor de Transacciones, que coordina las transacciones para los programas de aplicación.

Planificador (scheduler), que es el responsable de llevar a cabo el control de concurrencia.

Gestor de Recuperación, que se encarga de asegurar que la base de datos queda consistente después de algún fallo.

Gestor de Memoria Intermedia (cache, buffer), que se ocupa de la transferencia de los datos de memoria volátil a discos y viceversa.

2.5.2 El fichero Diario (LOG)

Para conseguir anular y recuperar transacciones, el método más extendido suele ser la utilización de un fichero denominado diario (log o journal) en el que se va guardando toda la información necesaria para deshacer – en el caso de fracasar- o rehacer si hay que recuperar las transacciones. Un registro de fichero diario suele constar de:

Identificador de la Transacción
Hora de modificaciones
Identificador del registro afectado
Tipo de acción
Valor anterior del registro
Nuevo valor del registro
Información adicional

Al irse almacenando todos los cambios en le fichero diario, puede surgir un problema en caso de que se realice un cambio en la BD y no en le fichero diario debido algún fallo del equipo; normalmente se obliga a que los registros que se modifican y que se encuentran en memoria de área intermedia o memoria principal, se escriban antes en el fichero diario que en la base de datos, para poder anular así, en caso de necesidad, las transacciones.

El fichero diario puede ser un fichero circular, es decir, que una vez lleno va eliminando registros según van entrando otros nuevos. Para evitar tener que recorrer todo el fichero diario, lo cual consumiría mucho tiempo, se introduce el concepto de punto de verificación o punto de recuperación (checkpoint), que se ejecuta periódicamente y que implica:

- Pasar el contenido de las memorias de área intermedia al fichero diario
- Escribir un registro de punto de recuperación en el diario
- Pasar el contenido de las memorias de área intermedia de la base de datos o soporte secundario
- Escribir la dirección del registro de recuperación en un fichero de rearranque

En el registro de punto de recuperación se reflejan todas las transacciones que se encuentran activas en el momento de producirse el punto de recuperación.

También existen otras formas de garantizar la recuperación ante fallos sin emplear el fichero diario, como es la técnica de paginas ocultas (shadow paging).

2.5.3 Recuperación en Caliente

Al ocurrir un fallo que de lugar a la pérdida de memoria volátil, es preciso realizar la operación que se suele denominar recuperación en caliente, en la que el sistema consulta el fichero diario para determinar las transacciones que hay que deshacer porque no han sido completadas y las que hay que rehacer porque, si bien se han completado, no habían sido grabadas en la base de datos cuando se produjo el fallo.

Con este fin, al recuperar la base de datos después de una caída del sistema, se obtiene la dirección del registro de recuperación más reciente y se recorre el fichero diario desde este punto hasta el final. Así por ejemplo, como muestra la figura 2.1, la transacción T1 no se vería afectada por el proceso de recuperación, las transacciones T2 y T4 deberían ser rechazadas y las transacciones T3 y T5 deberán deshacerse ya que no han concluido. Esto será así en caso de utilizar un protocolo de actualización inmediata, según el cual las actualizaciones se realizan en la base de datos a medida que se producen.

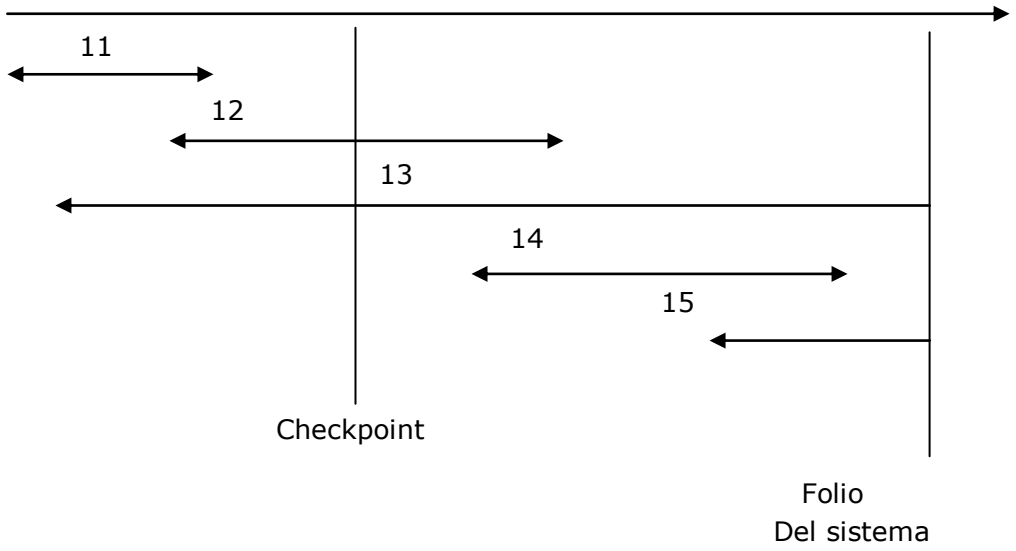


Figura 2.1 Recuperación en Caliente

Si se emplea el protocolo de actualización diferida, en el que las actualizaciones no se llevan a cabo en la base de datos hasta que finaliza, no es necesario deshacer transacciones.

2.5.4 Recuperación en Frío

En caso de un fallo de memoria secundaria que afecte la base de datos, se llevará a cabo una recuperación en frío, que consiste en utilizar una copia de seguridad de la BD, también llamada de respaldo (backup), que permitirá, junto con los ficheros diarios que se han ido produciendo desde que se realizo la copia de seguridad, reconstruir la base de datos llevándola de forma consistente a la situación anterior a que se produjera el fallo.

Otro caso que se puede dar es el denominado error fatal que se produce cuando se pierde el fichero diario grabado en un soporte, en este caso resulta imposible recuperar la base de datos a su estado actual. La mejor solución para evitar este problema es la que ofrecen algunos SGBD, que permiten la gestión de copias del fichero diario en dispositivos independientes. También se puede duplicar la base de datos. En general, todas las técnicas de duplicación se conocen como espejo (mirroring) o duplexación (duplexing).

2.5.5 Recuperación en Bases de Datos Distribuidas

Hay que destacar que los principales fallo en bases de datos distribuidas pueden deberse a:

-Fallos en los nodos, hablándose de fallo total cuando todos los nodos están caídos, y de fallo parcial si sólo se encuentran caídos algunos de ellos.

-Fallos en los enlaces de comunicación, debido a que haya ruido en la línea o ésta se corte. Puede incluso darse el caso en el que todos los caminos entre dos nodos se encuentren imposibilitados, produciéndose una partición en la red, esto es, quedando dividida en varios componentes.

-Mensajes no entregables, debido a que el nodo no es operativo, pudiéndole persistir en le intento de envirarlo o bien eliminarlo.

Los protocolos de red suelen proporcionar diversas técnicas para contrarrestar estos fallos como son: códigos de detección de errores, retransmisión de mensajes, re encaminamiento, copas de espera (time up), etc.

De todas maneras el SGBD debe preocuparse de asegurar la atomicidad de las transacciones, para lo que se han elaborado diversos protocolos de grabación atómica entre los que se destaca el de grabación en 2 fases o two-phase-comete. En este protocolo un componente se encarga de coordinar a los demás que participan en una transacción, actuando de la siguiente manera:

-primera fase: en coordinador pregunta a los participantes si están preparados para grabar la transacción.

-segunda fase: el coordinador recibe la respuesta de los participantes, tomando la decisión de grabar o deshacer la transacción y comunicándosela a todos los participantes.

Si un participante responde que no encuentra listo para grabar la transacción, el coordinador ordena a los demás que aborten la transacción. Solo si todos han contestado que están preparados para grabar, la transacción se graba.

Si un nodo (participante o coordinador) no recibe respuesta, se invoca un protocolo de terminación. Por otro lado, si un nodo se ha caído, se emplea un protocolo de recuperación, que terminará las acciones a realizar dependiendo del momento en el que el nodo haya fallado (por ejemplo, si un participante falla antes de la primera fase a recuperarse abortaría la transacción).

Hay que destacar que en un SGBD distribuido, además del gestor de transacciones local, existe un gestor de transacciones global en cada nodo, encargado de coordinar la ejecución de transacciones globales.

2.6 INTEGRIDAD

El objetivo en cuanto a la integridad es proteger la base de datos contra operaciones que introduzcan inconsistencia en los datos, por eso hablamos de integridad en el sentido de corrección, validez, o precisión de los datos de la base.

El subsistema de integridad de un SGBD debe, por tanto, detectar y corregir, en la medida posible las operaciones incorrectas. Hay que tener en cuenta, sin embargo que habrá operaciones cuya falta de corrección no sea detectable, por ejemplo si se introduce como fecha de nacimiento de un empleado el 17/02/31 cuando en realidad era el 19/02/31, este error nunca podrá ser detectado por el sistema ya que ambas fechas son igualmente válidas.

Existen dos tipos de operaciones que puedan atentar contra la integridad de los datos que son las operaciones semánticamente inconsistentes y las interferencias debidas a accesos concurrentes.

2.6.1 Integridad Semántica

Existen operaciones que pueden violar restricciones definidas al diseñar la base de datos, como pueden ser restricciones sobre los dominios (el estado civil tiene como valores soltero, casado, divorciad, viudo), o sobre os atributos (fecha de nacimiento debe ser tal que sea menor a los 70 años), estas restricciones pueden ser estáticas o dinámicas.

Los SGBD tienen que ofrecer en su lenguaje de definición facilidades que permitan describir las restricciones, con una sintaxis adecuada y gran flexibilidad. Lo más deseable es que esta definición se haga de forma declarativa, aunque una gran parte de los productos existentes exige emplear un enfoque procedimental (por medio de disparadores y procedimientos almacenados).

Un aspecto muy importante en estas reglas es que se almacena en un diccionario, como parte integrante de la descripción de los datos(control centralizado de la semántica), de modo que ya no han de incluirse en los programas, con lo que se consiguen las siguientes ventajas:

+Las reglas de integridad son mas sencillas de entender y de cambiar, facilitando su mantenimiento.

+Se detectan mejor las inconsistencias

+Se protege mejor la integridad, ya que ningún usuario podrá escribir un programa que las viole llevando la base de datos a estados inconsistentes.

Los subsistemas de integridad del SGBD deben realizar las siguientes funciones:

- Comprobar la coherencia de las reglas que se definen
- Controlar las distintas transacciones y detectar las violaciones de integridad
- Cuando se produce una violación, ejecutar las acciones pertinentes.

2.6.2 Integridad Operacional

El sistema multiusuario es imprescindible, además, un mecanismo de control de concurrencia para conservar la integridad de la base de datos, ya que se pueden producir importantes inconsistencias derivadas del acceso concurrente.

Aquí mencionaremos algunos problemas clásicos:

· Operación perdida
·Salidas inconsistentes
·Introducción de inconsistencia en la base de datos
·Lectura no reproducible

Para asegurar la consistencia de las transacciones se tiene que cumplir que sean seriables, en el sentido de que el efecto de ejecutar transacciones de forma concurrente debe ser el mismo del que se producirán al ejecutar por separado en un orden secuencial según van entrando al sistema. El concepto de seriabilidad es fundamental para el control de concurrencia.

De aquí se desprende nuestro siguiente subtema que son las técnicas clásicas del control de concurrencia.

2.6.3 Técnicas Clásicas

Bloqueo:

Se puede definir bloqueo como una variable asociada a cada elemento de datos, que describen el estado de dicho elemento respecto a las posibles operaciones (recuperando o actualizando) que se pueden realizar sobre ellos en cada momento. Las transacciones pueden llevar a cabo bloqueos, por ejemplo sobre los registros que vayan a utilizar, impidiendo a otros usuarios la

recuperación o actualización de los elementos bloqueados, pudiendo así evitar inconsistencia en le acceso concurrente.

Aunque el propio SGBD gestiona ciertos bloqueos a fin de asegurar la consistencia, también los usuarios pueden bloquear, de forma explicita, los objetos deseados, a fin de impedir la actualización o acceso por parte de otros usuarios.

Los bloqueos pueden ser de varios tipos:

Bloqueos exclusivos o de escritura: cuando una transacción mantiene un bloqueo de este tipo sobre un objeto, ninguna otra transacción puede acceder a el, ni adquirir ningún tipo de bloqueo sobre ese objeto, hasta que sea liberado.

Bloqueos compartidos o de lectura: permite que otras transacciones retengan también ese mismo objeto en bloqueos compartidos, pero no exclusivos.

El tema de los bloqueos es muy importante al afectar fuertemente al rendimiento del sistema. Los SGBD difieren muchas veces en los niveles del bloqueo:

- Un campo
- Un registro/tuna tupla
- Un fichero/una relación
- La base de datos en su totalidad

Se suelen utilizar bloqueos de varias granularidades, lo que puede hacer ineficiente el sistema.

MARCAS DE TIEMPO (timestamping)

Las marcas son identificadores únicos que se asignan a las transacciones y que pueden considerarse como el tiempo de inicio de la transacción.

Esta técnica permite ordenar las transacciones y controlar un acceso en secuencia de la misma a los datos.

Existe varios protocolos basados en marcas de tiempo, entre los que destacan: WAIT-DIE fuerza a una transacción a esperar; WOUND-WAIT permite a una transacción matar a otra.

MARCAS DE TIEMPO MULTIVERSION

El mecanismo de marcas de tiempo supone que existe una única versión de los datos por lo que solo una transacción puede acceder a los mismos. La existencia de múltiples versiones elimina la necesidad de sincronización entre distintas operaciones de escritura ya que cada una produce una nueva versión y no entra en conflicto con otra.

TECNICAS OPTIMISTAS

Permiten que las transacciones accedan libremente a los objetos, determinando antes de su finalización si ha habido o no indiferencias. Cada transacción consta de 2 o 3 frases: una frase de lectura, una de validación, y posiblemente, una de escritura.

2.6.4 Aspectos Avanzados

Las aplicaciones avanzadas de bases de datos, que soportan sistemas CAD/CAM, CASE, OIS, GIS, etc. Requieren nuevos mecanismos de control de concurrencia que permiten anidar transacciones y que faciliten la coordinación entre varios usuarios.

TRANSACCIONES ANIDADAS:

Es como una composición de un conjunto de subtransacciones que a su vez pueden ser anidadas. Las subtransacciones de una transacción se pueden ejecutar concurrentemente, pudiendo cancelarse o reiniciarse una subtransaccion sin afectar a la transacción de la que forma parte.

Esta técnica no alerta la serialidad, pero si mejora el rendimiento, lo que puede ser muy importante para una recuperación rápida de la base de datos.

TRANSACCIONES LARGAS:

Existen diferentes propuestas que pueden soportar la larga duración como son:

Las que extienden técnicas basadas en la serialidad: bloqueos altruistas, validación por medio de instancias, transacciones multinivel.

Las que relajan la serialidad, utilizando semánticas de datos o semántica especifica de aplicación, sagas, corrección de predicados conflictivos, restauración dinámica de transacciones, etc. Aquí está el bloqueo altruista y el control de concurrencia basado en semántica.

CAPITULO 3. CRIPTOGRAFIA

3.1 ANTECEDENTES

Se puede decir que la criptografía es tan antigua como la civilización; cuestiones militares, religiosas o comerciales impulsaron desde tiempos remotos el uso de escrituras secretas; los antiguos egipcios usaron métodos criptográficos, mientras el pueblo utilizaba la lengua demótica, los sacerdotes utilizaban la escritura hierática (jeroglífica) incomprensible para el resto. Los antiguos babilonios también utilizaron métodos criptográficos en su estructura cuneiforme. El primer caso claro de uso de métodos criptográficos se dio durante la guerra en Atenas y Esparta, el cifrado se basaba en la alteración del mensaje original mediante la inclusión de símbolos innecesarios que desaparecerían al enrollar la lista en un rodillo llamado hesítala, el mensaje quedaba claro cuando se enrollaba la tira de papel alrededor del rodillo de longitud y grosor adecuados. Carlomagno sustituía ya las letras por símbolos extraños. En la época de los romanos se utilizó el cifrado Cesar que consistía en cambiar cada letra por la que ocupaba tres lugares mas adelante en el abecedario.

En la edad media San Bernardino evitaba la regularidad de los signos (por lo que el criptoanálisis por el método de las frecuencias no era efectivo) sustituyendo letras por varios signos distintos, así tenia un símbolo para cada consonante, usaba tres signos distintos para cada una de las vocales y utilizaba signos sin ningún valor. El libro más antiguo del que se tiene constancia y que trata de criptografía es el Liber Zifrorum escrito por Cicco Simoneta en el siglo XIV. En el siglo XV destaca Leon Battista Alberti que es considerado por muchos el padre de la criptografía; crea la primera máquina de criptografiar que consiste en dos discos concéntricos que giran independientes consiguiendo con cada giro un alfabeto de trasposición. En el siglo XVI, Girolamo Cardano utilizo el método de la tarjeta de agujeros perforados, que se debía colocar sobre un texto para poder leer el mensaje cifrado; en este mismo siglo Felipe II utilizó una complicada clave que en francés Viete logro descifrar.

Carlos I de Inglaterra usó en el siglo XVII códigos de sustitución silábica. Napoleón, en sus campañas militares y en los escritos diplomáticos, usó los llamados métodos

Richelieu y Rossignol y para evitar la regularidad de los símbolos asignaba números a grupos de una o más letras.

En el siglo XIX se utiliza ampliamente el método de transposición, consistente en la reordenación según distintos criterios de los símbolos del mensaje. Kerckhoffs escribe el libro La Criptografía Militar en el que da las reglas que pueden cumplir un buen sistema criptográfico. El al primera guerra mundial los alemanes utilizaron el sistema denominado ADFGX, en el que a cada combinación de dos letras del grupo ADFGX se le hace corresponder una letra del alfabeto y a la que posteriormente se le hacia una transposición en bloques de longitud 20. El presidente americano Jefferson diseñó un cilindro formado por varios discos que se utilizaba como máquina criptográfica. El mayor desarrollo de la criptografía se dio en el periódico de entreguerras por la necesidad de establecer comunicaciones militares y diplomáticas seguras.

En 1940 se construyó la máquina Hegelin C-48, consistente en 6 volantes unidos por el eje y con distinto número de clientes. En la segunda guerra mundial se construyó por parte alemana la máquina Enigma, que se basaba en un perfeccionamiento del cilindro de Jefferson, pero la máquina británica Colossus consiguió descifrar los mensajes cifrados con Enigma. Los americanos construyeron la maquina Magic utilizada para descifra el código purpura japonés; los americanos a su vez usaron a los indios navajeros con su difícil mensaje para la transmisión de mensajes.

Con el desarrollo de la informática en la segunda mitad de este siglo y con el uso más extendido de las redes informáticas y del almacenamiento masivo de información se ha dado paso a un gran salto en el estudio de sistemas criptográficos. En 1975 Diffie y Hellman establecieron las bases teóricas de los algoritmos de llave pública, hasta entonces no se concebía un sistema de cifrado que no fuese de clave secreta. En la actualidad se usan distintos métodos criptográficos, el DES (de llave secreta), método RSA, método de Merkie y Hellman, etc.

3.2 CONCEPTOS BASICOS

En la práctica, la criptografía se ocupa del cifrado y el descifrado de mensajes. Cifrar información consiste en transformar un mensaje en claro (plaintex) en un mensaje cifrado mediante el uso de una clave.

El texto en claro es inteligible, mientras el texto cifrado es ininteligible, es decir, tiende a parecerse a una sucesión de caracteres aleatorios a los que no es posible otorgar ningún significado.

La operación inversa al cifrado es el descifrado, y consiste en obtener el texto en claro a partir del texto cifrado.

Es necesario hacer una distinción entre código y cifra, con un código se sustituye una palabra o frase del texto original por otra palabra o frase del texto cifrado. La traducción a una lengua extranjera es un buen ejemplo de código, así podemos transformar el mensaje original "estoy contento" por "carita feliz" o "emprender la guerra" por "to go to war". La cifra suele actuar antes sobre caracteres que sobre palabras, utilizando un sistema de signos en el que se transcriben guarismos, letras o símbolos según una clave acordada; es posible, por ejemplo, cifrara el mensaje

"emprender la guerra" escribiendo "merpneedlrgaerra" donde la clave utilizada ha sido: eliminar los espacios en blanco, tomar las letras de dos en dos y cambiar su orden escribiéndola.

COMPONENTES DE UN CRIPTOSISTEMA

Todo sistema criptográfico o criptosistema consta de 5 componentes básicos:

1.- El espacio de mensajes, que es el conjunto de los posibles textos en claro. Los elementos de este conjunto se denominan mensajes, teniendo en cuenta que nos referimos a mensajes inteligibles. Los mensajes se forman a partir de un alfabeto, mediante unas reglas sintácticas del idioma en que se origina.

2.- El espacio de cifrado o de texto cifrados, es el conjunto de todos los posibles mensajes cifrados. El alfabeto de los textos cifrados puede ser el mismo o ser distinto del utilizado para los mensajes en claro.

3.- El espacio de las claves, es el conjunto de las posibles claves utilizadas en los procesos de cifrado y descifrado.

4.- Una familia de transformaciones de cifrado, donde un parámetro, denominado clave de cifrado, define la transformación concreta realizada.

5.- Una familia de transformaciones de descifrado, donde la clave de descifrado define la transformación utilizada.

METODOS CRIPTOGRAFICOS BÁSICOS

SUSTITUCION:

Este método consiste básicamente en sustituir los caracteres del mensaje inicial por otros; los nuevos caracteres pueden ser de cualquier tipo: letras, símbolos, dígitos, etc.… Se pueden considerar dos tipos de sustitución:

1) Equivalencia entre alfabetos carácter a carácter. A cada letra del alfabeto ordinario se le hace corresponder un símbolo y el mensaje se cifra cambiando las letras iníciales por su equivalente. Un ejemplo; la letra A se le asigna el símbolo "@" en el mensaje cifrado siempre tendremos el símbolo @ en vez de la letra A.

2) Utilización de cifra o clave. Una vez establecida la correspondencia entre alfabetos, la asignación de caracteres se realiza teniendo en cuenta la posición del carácter en el mensaje y el digito que le corresponde según su clave. Ejemplo: sea le mensaje "SECRETO" y la cifra "23" el mensaje cifrado se consigue adelantando 2 letras la primera que se encuentre, 3 la segunda, 2 la tercera, 3 la cuarta y así sucesivamente, así el mensaje será: "UHEUGWQ", la letra e se ve ahora como h y otra como g entonces ya no hay correspondencia uno a uno entre el alfabeto inicial y los símbolos del mensaje cifrado. Este método se llama Vigenere.

TRANSPOSICION.

Este consiste en una reordenación de los símbolos del mensaje original de modo que este resulte ilegible. Si un mensaje consta de n letras se podrá transponer de n! formas. La reordenación se puede realizar desde un modo simple: escribiendo el

mensaje letra a letra pero al revés, o utilizando esquemas matriciales. Los métodos de sustitución y transposición se pueden combinar para formar métodos mixtos más seguros ante un ataque criptográfico.

METODOS DE LAS FRECUENCIAS.

Este método consiste en usar la permanencia estadística de los símbolos utilizados después de la sustitución; así si el símbolo $ es el que más aparece en el criptograma, es posible pensar que con bastante probabilidad dicho símbolo corresponderá a alguna vocal.

Por lo tanto cuando se tiene un criptograma lo suficientemente largo y se sospecha que está cifrado por sustitución simple, el primer paso del criptoanálisis debe ser realizar una tabla con las frecuencias de cada uno de los símbolos, ordenar dichas tablas de mayor a menor; después se deberán asociar los símbolos con los correspondientes a los que aparecen las letras en español, dicha tabla se puede componer a partir de un texto cualquiera lo suficientemente largo.

Un criptosistema sencillo

A continuación se muestra un ejemplo de criptosistema. Se denomina cifrado Cesar y fue utilizado por Julio Cesar en sus campañas militares. Se trata de un sistema de sustitución simple en el que cada letra del alfabeto del mensaje es sustituida por una letra situada 3 posiciones mas adelante para obtener el mensaje cifrado. Por ejemplo, la letra A es sustituida por la D, la B por la E, la C por la F y así sucesivamente, como se muestra en la siguiente figura 3.1:

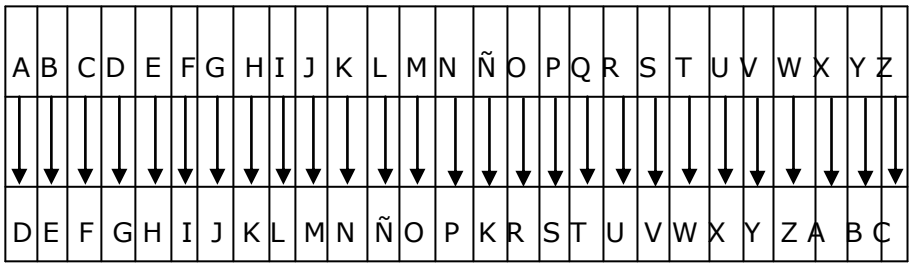


Figura 3.1

En la figura de muestra gráficamente como quedaría la sustitución de las letras.

En este caso, la transformación de cifrado consiste en sustituir cada carácter por uno situado K posiciones mas adelante. No solo eso sino que la clave k está fijada con un valor 3 en el caso del cifrado César.

La transformación de cifrado es muy sencilla, ya que consistente en sustituir cada carácter por un situado k posiciones por detrás. Como vemos se cumple que la transformación de descifrado es la inversa del mensaje cifrado.

Base teórica de la criptografía

La criptografía descansa sobre tres importantes campos teóricos:

- La teoría de la información
- La teoría de los números
- La teoría de la complejidad algorítmica

La fundamentación matemática de la teoría de la comunicación y su posterior aplicación a los sistemas criptográficos pueden encontrarse en los trabajos de C.E Shannon. A partir de las investigaciones de este autor, la criptografía deja de ser un arte y pasa a convertirse en una ciencia.

La seguridad de los sistemas criptográficos descansa sobre dos conceptos fundamentales, como son la difusión y la confusión.

El propósito de la difusión de la información, es distribuir las propiedades estadísticas de los mensajes en claro, sobre todo el texto cifrado.

El propósito de la confusión es establecer una relación lo más compleja posible entre la clave y el texto cifrado. De este modo un criptoanalista no podrá deducir información acerca de la clave mediante un estudio de texto cifrado.

Ambas técnicas, la difusión y la confusión, por separado, proporcionan fortaleza a los criptosistemas, sin embargo utilizadas conjuntamente pueden dar lugar a sistemas muy difíciles de atacar. Un ejemplo de combinación de estas técnicas es el DES (Data Encryption Stándard) en éste las permutaciones proporcionan difusión y las sustituciones la confusión.

Clasificación de los Criptosistemas

Restringidos: basan su técnica en mantener secreta la naturaleza del cifrado y del descifrado.

Clave Privada: basa su técnica en un valor secreto, llamado clave.

Clave Públicas: basa su técnica en que la clave para cifrar es pública, mientras que la de descifrar solo es conocida por el usuario correspondiente, y además es computacionalmente difícil encontrar la clave de descifrado a partir del conocimiento de la clave de cifrado.

Cuánticos: se basa en aspectos de la física cuántica

Probabilísticos: basan su técnica en que cifrar el mismo mensaje por la misma clave no siempre da el mismo mensaje cifrado.

Requisitos de un criptosistema

Existen tres técnicas fundamentales utilizadas por los criptoanalistas para atacar un criptosistema, aunque casi siempre se utilizan combinaciones de ellas. En general las modernas técnicas de criptoanálisis suponen conocimientos matemáticos avanzados y utilizan mecanismos estadísticos, software y hardware muy sofisticados y en ocasiones muy caros.

1. Ataque a partir solo de texto cifrado
2. Ataque a partir de un mensaje conocido
3. Ataque por elección de mensaje

3.3 CRIPTOGRAFÍA EN LA SEGURIDAD INFORMÁTICA

La criptografía puede aplicarse en dos ámbitos de la seguridad informática: en el almacenamiento de la información y en la transmisión de la misma.

La criptografía es fundamental para la seguridad. Aunque se superen las barreras de seguridad física establecidas, e incluso las barreras de seguridad lógica para el control de accesos en el sistema operativo, la criptografía permite mantener algunas de las características de la seguridad informática.

Aunque no salvaguarda la integridad de los datos ante un posible borrado total o parcial de los mismo, si asegura su integridad en el sentido que le facilita la detección de cualquier tipo de modificación, incluido el añadido o barrado de información. En primera instancia protege el secreto/confidencialidad de la información.

Cabe señalar que la criptografía no puede utilizarse para garantizar la disponibilidad de la información. Esta característica debe ser preservada mediante el uso de otro tipo de mecanismos.

SECRETO O CONFIFENCIALIDAD

Está claro que el cifrado de la información es un excelente método para proteger su confidencialidad. Aunque se acceda a la información, o se intercepte mientras se transfiere, si está cifrada sigue siendo inútil a menos que se descifre.

INTEGRIDAD Y PRESICION

Algunos sistemas criptográficos incorporan medios para prevenir que se dañe la integridad de la información, esto es, que ésta sea modificada voluntaria o involuntariamente. El sistema permite detectar cualquier pequeño cambio que se haya producido en el mensaje original.

AUTENTICACION

La criptografía también puede usarse para asegurar la autenticidad de los mensajes. Esto es, asegurar que el mensaje a sido enviado por quien se identifica como su emisor. Se trata de identificar sin posible error el origen de los mensajes.

En relación con la autenticación suelen utilizarse las denominadas firmas digitales. Se trata de añadir algún tipo de información en el mensaje o de utilizar de algún modo las claves para validar al destino el origen del mensaje.

En el ámbito de la trasferencia de mensajes cifrados de autenticación está muy relacionada con la integridad. Así, la autenticación influiría tres aspectos:

1.- Asegurar que el mensaje no ha sido alterado, ni maliciosa ni descuidadamente durante su transmisión. El mensaje llego tal y como se envío (integridad).

2.- Asegurar que el mensaje no es el reenvío de uno previamente emitido e interceptado (no reenvío).

3.- Asegurar que el emisor es quien dice que es (autenticidad).

NO REPUDIO

Es una característica que se relaciona con la transmisión de mensajes cifrados. Se trata de prevenir que la persona que envió el mensaje pueda alegar con posterioridad que el no envío ese mensaje (repudiarlo). El receptor debe disponer de mecanismos que demuestren ante terceros que solo el emisor pudo enviar el mensaje.

3.4 SISTEMAS DE CIFRADO CLÁSICOS

Se consideran criptosistemas clásicos aquellos que son anteriores al uso sistemático de las computadoras en el campo de la criptografía. Sus características fundamentales son su simplicidad y la facilidad para recordar los algoritmos y la clave.

Dado que se aplicaban en el ámbito militar los mensajes tenían que poder cifrarse y descifrarse de modo rápido y sencillo, y el método utilizado debería ser fácil de recordar. Estas características convertían a los sistemas en muy débiles y fáciles de atacar mediante métodos muy sencillos.

Fundamentalmente, se pueden distinguir dos tipos de cifrado clásicos:

Por transposición y por sustitución.

CIFRADOS POR TRANSPOSICION O PERMUTACION

Se reordenan los bits, caracteres o bloques de caracteres de texto en claro para obtener el texto cifrado.

El método más sencillo de este tipo de sistemas es el método de transposición simple, el cual consiste en una reordenación de los símbolos del mensaje original, de modo que este resulte ilegible. Simplemente desordenando las unidades que forman el texto original según la clave, dividiéndose el mensaje original en bloques de longitud n y aplicándose a cada bloque la transposición determinada por la clave elegida.

Ejemplo

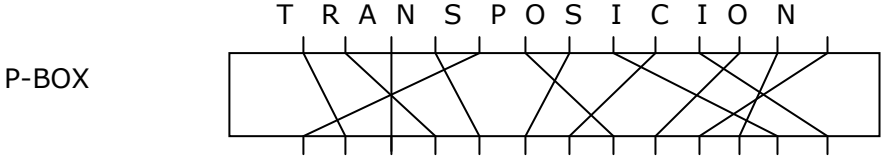


Figura 3.2 Esta figura muestra como quedaría el intercambio de las letras y sus posiciones.

Cuando este sistema se aplica en computadoras, el dispositivo encargado de permutar cada bloque de texto suele denominarse P-box (permutación box).

Otro ejemplo es el denominado transposición por columnas. En este se dispone el texto por filas de una determinada longitud, rellenándose al final de la fila por un carácter cualquiera. El texto cifrado se obtiene leyendo la matriz por columnas. La clave de descifrado es simplemente el número de columnas utilizado.

Ejemplo

La frase EN UN LUGAR DE LA MANCHA

E	N	U
N	L	U
G	A	R
D	E	L
A	M	A
N	C	H
A	X	X

El texto cifrado seria: ENGDANANLAEMCXUURLAHX

CRIFRADOS POR SUSTITUCION

Se remplazan bits, caracteres o bloques del texto en claro por otros en el texto cifrado. La versión más sencilla de este método es el denominado cifrado por sustitución simple o monoalfabeto. En este cada carácter del texto en claro es siempre sustituido por un mismo carácter en el texto cifrado.

Un caso de sustitución simple es el cifrado Cesar, en el que, como ya se vio, cada carácter es sustituido por el situado tres posiciones adelante en el alfabeto.

Otra modalidad del sistema por sustitución es el cifrado por sustitución polialfabeto. En este, cada carácter del texto en claro es sustituido por un carácter distinto en le texto cifrado cada vez que aparece.

Un ejemplo

Del sistema cifrado polialfabeto es el método de Vigenere. En este se utiliza como clave una palabra cuyas letras definen el desplazamiento de los distintos alfabetos a usar.

Supongamos que se utiliza como palabra clave SOL se tendría lo siguiente:

A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

Alfabeto 1

S T U V W X Y Z A B C D E F G H I J K L M N Ñ O P Q R

Alfabeto 2

O P Q R S T U V W X Y Z A B C D E F G H I J K L M N Ñ

Alfabeto 3

L M N Ñ O P Q R S T U V W X Y Z A B C D E F G H I J K

Cifrado

Mensaje P L A N T A A T O M I C A

Clave S O L S O L S O L S O L S

Cifrado I Z L F I L S I Z E W N S

Sustitución y Transposición no resultan muy efectivos usados individualmente, sin embargo constituyen la base de sistemas más difíciles de criptoanalizar.

3.5 SISTEMAS DE CIFRADO MODERNOS

Los sistemas criptográficos modernos se desarrollan con la aparición de las computadoras, y basan su funcionamiento en la utilización de potentes y complejas herramientas hardware y software. Se utilizan claves secretas de gran longitud para controlar una compleja secuencia de operaciones con la información, que pueden incluir tanto transposiciones como sustituciones. Su posibilidad de uso se basa en la potencia y en la capacidad de las maquinas, que permiten aplicar algoritmos de gran complejidad y costos en tiempo admisibles.

Los criptosistemas modernos pueden dividirse en dos grandes categorías en función del tipo y numero de claves que utilizan:

- Criptosistemas simétricos, también llamados de clave única o de clave privada.
- Criptosistemas asimétricos, también llamados de clave pública o de dos claves.

Ambos tipos de sistemas suelen combinarse para llevar acabo distintas acciones y lograr ciertos objetivos de seguridad.

Criptosistemas de Clave Privada

En estos sistemas se utiliza la misma clave para el cifrado y descifrado. Esta clave se denomina clave privada(secreta o única) debido a que solo es conocida por el emisor y por el receptor del mensaje. Para que este tipo de sistemas sea efectivo la clave debe ser mantenida en secreto por ambos componentes de la comunicación.

La seguridad de este tipo de sistemas depende totalmente del nivel de protección de la clave. Cuando se descifra un mensaje usando al clave privada, el hecho de que esta sea tan solo conocida por el emisor y el receptor garantiza dos propiedades:

1. Que el mensaje no es inteligible para nadie más, es decir, que es confidencial.
2. Que si el texto descifrado es inteligible, sólo hay un emisor posible, aquel que conoce la clave privada. Esto asegura la autenticidad del mensaje.

Por lo tanto este tipo de sistema, el secreto (confidencialidad) y la autenticidad se obtiene al mismo tiempo. Las claves privadas deben intercambiarse de modo totalmente seguro, pues sobre ellas descansan todas las características de seguridad del sistema.

Criptosistema de Clave Pública

En este tipo se sistema se utilizan 2 claves: una clave pública y una clave privada. En un grupo de usuarios, cada uno de ellos posee dos claves distintas:

La clave pública, K´, como su propio nombre indica, puede ser conocida por todos los usuarios del sistema.

La clave privada, K, que tan solo es conocida por el propietario.

Aunque estas claves estén relacionadas matemáticamente, la fortaleza del sistema depende de la imposibilidad computacional de obtener una a partir de la otra.

Este tipo de sistemas de denominan asimétricos por que no es posible usar una misma clave para cifrar y descifrar un mensaje. Ambas claves deben usarse en el proceso. Si se cifra un mensaje con una de ellas, se deben descifrar con otra.

Si un usuario (emisor) quiere enviar un mensaje secreto a otro (receptor), debe cifrarlo utilizando la clave pública del receptor.

El mensaje tan solo puede descifrarse utilizando la clave privada del receptor, con lo que se garantiza la confidencialidad del mismo. La clave pública del receptor no sirve para descifrar el mensaje, y por tanto tan solo el receptor puede descifrarlo.

Por otro lado el mensaje no es autentico. Dado que cualquier usuario puede conocer la clave pública del receptor, cualquier usuario puede ser el emisor del mensaje. La recepción de un mensaje cifrado con la clave pública del receptor no identifica unívocamente al emisor, y por lo tanto no lo autentifica.

El receptor podrá descifrarlo usando la clave pública del emisor. Dado que todo el mundo puede conocer la clave pública del emisor, no se garantiza la confidencialidad del mensaje. Cualquiera puede descifrarlo. Sin embargo, dado que solo el emisor conoce la clave privada, tan solo él puede saber el origen del mensaje, con lo que se garantiza la autenticidad del mismo. En este tipo de sistemas, el secreto y la autenticidad del mensaje se obtienen por separado. Para lograr las dos características de seguridad es necesario combinar ambas claves y realizar un doble proceso de cifrado y descifrado.

Criptosistemas Híbridos

Tanto la criptografía de clave pública como la de la clave privada tienen sus ventajas y sus inconvenientes. Debido a ello se suelen utilizar para distintos fines, y por lo tanto los criptosistemas de claves públicas no son un sustituto de las claves privadas.

Existen 2 razones que hacen que la criptografía de clave publica sea poco adecuada para la transferencia de información cifrada:

1.- Los algoritmos de la clave pública son lentos.

2.- Los algoritmos de la clave pública son vulnerables ataques mediante elección de mensaje.

Debido a estas razones la transferencia de información cifrada se suelen realizar mediante criptosistemas de clave privada, mientras los de la clave pública se reservan para funciones tales como la transferencia de claves. Lo ideal es combinar ambos tipos de criptosistemas para lograr una transmisión segura.

3.6 EL SISTEMA DES Y SUS MODOS

Origen del DES

El DES (Data Encryption Standard) es uno de los sistemas de cifrado de uso más extendido, dado que se ha convertido en un estándar reconocido por las agencias americanas y que se trata de un sistema de gran fortaleza.

El origen del DES se basa en una petición realizada en 1973 por el NBS (National Bureau of Standards) a distintos fabricantes para someter criptosistemas que pudieran servir como base a un estándar de cifrado de textos reservados no clasificados.

IBM disponía de un sistema altamente seguro denominado LUCIFER basado en una clave de 128 bits. Este sistema fue sometido al examen de la NBS y tras ser analizado por expertos de la NSA (National Security Agency) y ser reducido a 56 bits, fue aceptado y denominado DES.

Funcionamiento del DES

El DES se basa en la permutación de combinaciones y sustituciones realizadas sobre bloques de 64 bits de datos usando una clave de 56 bits.

La información a cifrar se divide en bloques de 64 bits, y sobre cada uno de ellos se repite el mismo proceso. Inicialmente se divide cada bloque en dos de 32 bits, Lo y Ro, y se permutan estos. Posteriormente se aplican 16 etapas en las que se combina cada bloque Li (aplicando la función XOR), producido en la etapa anterior, con el resultado de aplicar una función de bloque Ri en base a 48 bits de la clave inicial, Ki+1, dando lugar al bloque Ri+1.

En una última etapa, se deshace la permutación inicial reuniendo a los dos bloques resultantes para dar lugar a la salida cifrada. Por lo tanto el DES es reversible, es decir, puede aplicarse el mismo proceso para el cifrado como para el descifrado. Además pueden utilizarse las mismas claves para realizar ambos procesos, lo que lo convierte en un proceso simétrico. La clave K da lugar a 16 claves de 48 bits que se utilizan en cada unas de las 16 etapas del método. Si para el proceso de cifrado estas claves se utilizan en un orden, para el descifrado deben utilizarse en el orden contrario.

Con el fin de reforzar la seguridad de este sistema se han propuesto diversas modificaciones del mismo, entre las que se pueden destacar su aplicación reiterada (triple DES), o la ampliación de la longitud de sus claves.

Modos de DES

El criptosistema DES puede utilizarse en cuatro modos distintos en función de que se quieran obtener ciertas características, tales como poder transmitir a través de canales con ruido, autentificar el mensaje resultante, poder descifrar solo una parte del mismo, etc.

Los cuatro modos de operación del DES son:

- ECB (Electronic Code Book)
- CBC (Cipher Block Chainig)
- CFB (Cipher Feedback)
- OFP (Output Feedback)

Modo ECB (Electronic Code Book)

En modo ECB en claro es dividido en bloques de 64 bits que se cifran uno a uno y por separado usando el DES. La concatenación de los bloques cifrados da lugar al texto cifrado.

Este modo tiene el inconveniente de que es susceptible a ataques estadísticos y/o ataques sobre la clave, con un texto original conocido, sobretodo cuando las cabeceras de los textos tiene un formato estándar. Además se presenta el problema de que pueden eliminarse porciones de texto cifrado sin que se note, esto es, puede ocurrir que si se conocen las características y posición de cierta información en el texto en claro, esta puede eliminarse del texto cifrado sin impedir un correcto descifrado del mismo. Por otro lado este modo de funcionamiento tiene la ventaja de que trabaja bien en canales con ruido. Un fallo en la transmisión tan solo afecta a un bloque de 64 bits, no al mensaje completo. Este cifrado suele utilizarse para el cifrado de claves.

Modo CBC (Cipher Block Chaining)

En este modo, antes de cifrar cada bloque de 64 bits, se le aplica una XOR sobre el bloque cifrado anterior. El primer bloque se combina con un valor conocido. De este modo, se produce un encadenamiento (chaining) entre los distintos bloques, y el resultado de cifrar cada uno de ellos depende de todos los anteriores.

Debido a esta última característica, el último bloque del texto cifrado puede actuar como firma digital o checksum del resto, permitiendo certificar que no ha sido alterado. En este modo de funcionamiento, un error en el texto cifrado tan solo afecta al descifrado de dos bloques; es decir, tiene la ventaja de que impide la supresión y/o inserción de bloques de texto cifrado, puesto que en cualquier caso, el receptor sería incapaz de descifrar el criptograma recibido y, por lo tanto, quedaría alertado de las posibles intrusiones. Los ataques estadísticos, también se complican, debido a la interdependencia del texto cifrado a lo largo de todo el proceso. Este se suele utilizar para cifrar y autentificar documentos.

Modo CFB (Cipher Feedback)

En este modo se mantiene una cola de caracteres. Se cifran bloques sucesivos de 64 bits de la cola. El byte más significativo del resultado se combina (XOR) con el siguiente byte en claro para dar lugar al byte cifrado a transmitir. Además este último byte se reintroduce en la cola provocando un desplazamiento de su contenido. En este modo se utilizan para la seguridad de mensajes muy repetitivos y para cifrar/descifrar archivos, donde no es conveniente el almacén de información inútil.

Este método permite descifrar cualquier parte del texto cifrado sin conocer el resto. Además, realiza el cifrado a nivel de carácter, de tal manera que el texto cifrado va surgiendo de forma continua (stream mode) y no por bloques.

Modo OFB (Output Feedback)

Funciona de modo análogo al CFB, pero el byte realimentado en la cola es directamente el más significativo del cifrado de la misma.

Dado que un error en un bite de texto cifrado tan solo afecta a un bit en el texto descifrado, este suele utilizarse para comunicaciones satelitales.

EL SISTEMA RSA

El algoritmo de cifrado RSA, es el criptosistema de clave pública más extendido. Su nombre proviene de sus creadores Rivest, Shamir y Adleman, quienes lo desarrollaron en 1978.

Este sistema usa dos claves y cualquiera de las dos puede ser pública o privada. Las dos claves se generan matemáticamente basándose en parte en la combinación de grandes números con factores primos.

Desde su aparición, el sistema de llave publica RSA ha ganado gran popularidad, por una parte, por la gran seguridad que ofrece al basar ésta en un problema matemático difícil de resolver que había dejado interés en la comunidad mundial, como lo es el problema de la factorización entera (PFE), y a causa del sistema RSA, se ha retomado e incrementado su investigación. La seguridad del sistema depende de que las claves sean de enorme longitud y que una no pueda deducirse de la otra en un tiempo admisible. Dado que las claves se generan a partir del producto de dos números primos, la única forma de atacarla sería factorizandola en dichos números. Lo cual es demasiado costoso y complicado, ya que los números productos de 2 primos, son de los más difíciles de factorizar.

El sistema RSA, ha sido uno de los más estudiados hasta el momento y por lo tanto se considera que es uno de los más seguros, ya que se ha podido superar algunas controversias, así, actualmente es uno de los sistemas criptográficos de llave publica más usados en la industria, en el comercio, en le gobierno, en la milicia, en todas las actividades que requieran que la información tenga un alto grado de seguridad criptográfica. Solo los sistemas basados en el problema de logaritmo discreto elíptico (PLDE), estos criptosistemas públicos de curvas elípticas han podido competir con el sistema RSA.

CAPITULO 4. METODOLOGIA DEL ALGORITMO 'DES'

4.1 INTRODUCCIÓN

DES (*Data Encryption Standard*, estándar de cifrado de datos) es un algoritmo desarrollado originalmente por IBM a requerimiento del NBS (National Bureau of Standards, Oficina Nacional de Estandarización, en la actualidad denominado NIST, National Institute of Standards and Technology, Instituto Nacional de Estandarización y Tecnología) de EE.UU. y posteriormente modificado y adoptado por el gobierno de EE.UU. en 1977 como estándar descifrado de todas las informaciones sensibles no clasificadas. Posteriormente, en 1980, el NIST estandarizó los diferentes modos de operación del algoritmo. Es el más estudiado y utilizado de los algoritmos de clave simétrica.

El nombre original del algoritmo, tal como lo denominó IBM, era Lucifer. Trabajaba sobre bloques de 128 bits, teniendo la clave igual longitud. Se basaba en operaciones lógicas booleanas y podía ser implementado fácilmente, tanto en software como en hardware. Tras las modificaciones introducidas por el NBS, consistentes básicamente en la reducción de la longitud de clave y de los bloques, DES cifra bloques de 64 bits, mediante permutación y sustitución y usando una clave de 64 bits, de los que 8 son de paridad (esto es, en realidad usa 56 bits), produciendo así 64 bits cifrados

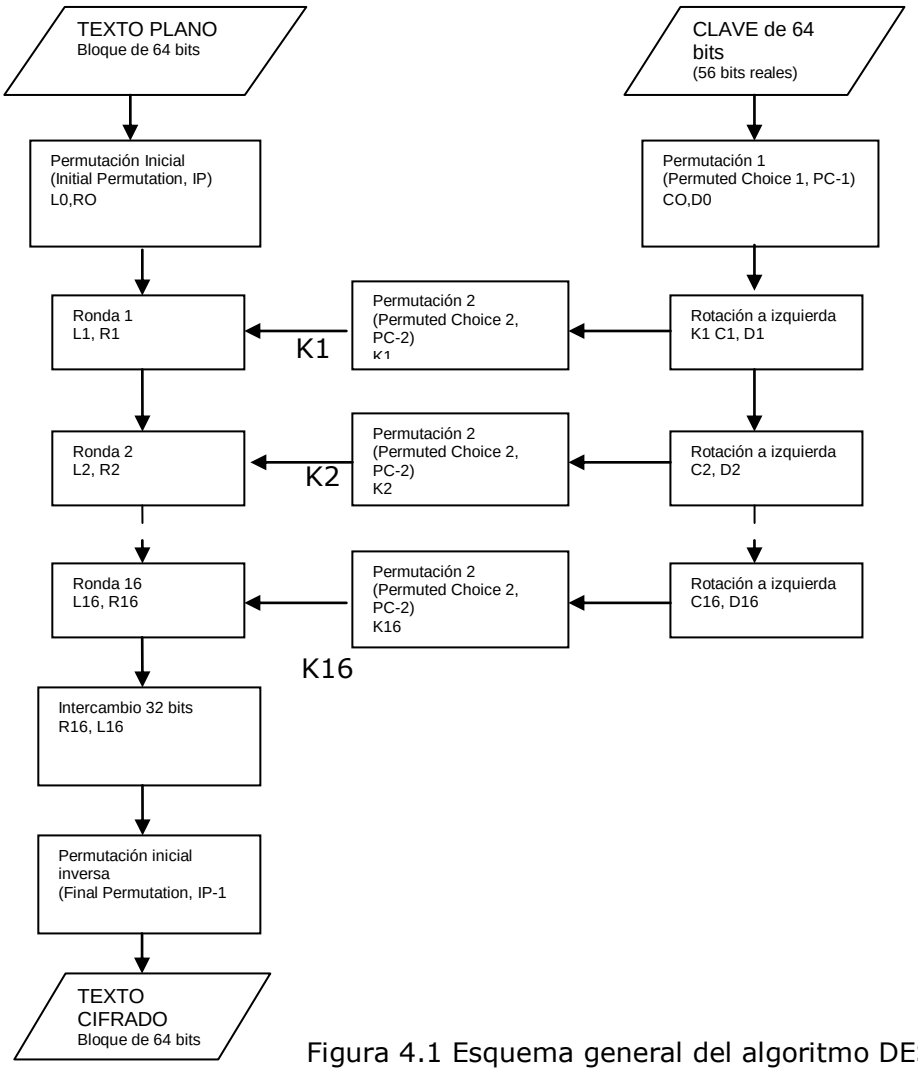


Figura 4.1 Esquema general del algoritmo DES

DES tiene 19 etapas diferentes.
La primera etapa es una transposición, una permutación inicial (IP) del texto plano de 64 bits, independientemente de la clave. La última etapa es otra transposición (IP-1), exactamente la inversa de la primera. La penúltima etapa intercambia los 32 bits de la izquierda y los 32 de la derecha. Las 16 etapas restantes son una Red de Feistel de 16 rondas.
En cada una de las 16 iteraciones se emplea un valor, K_i, obtenido a partir de la clave de 56 bits y distinto en cada iteración.

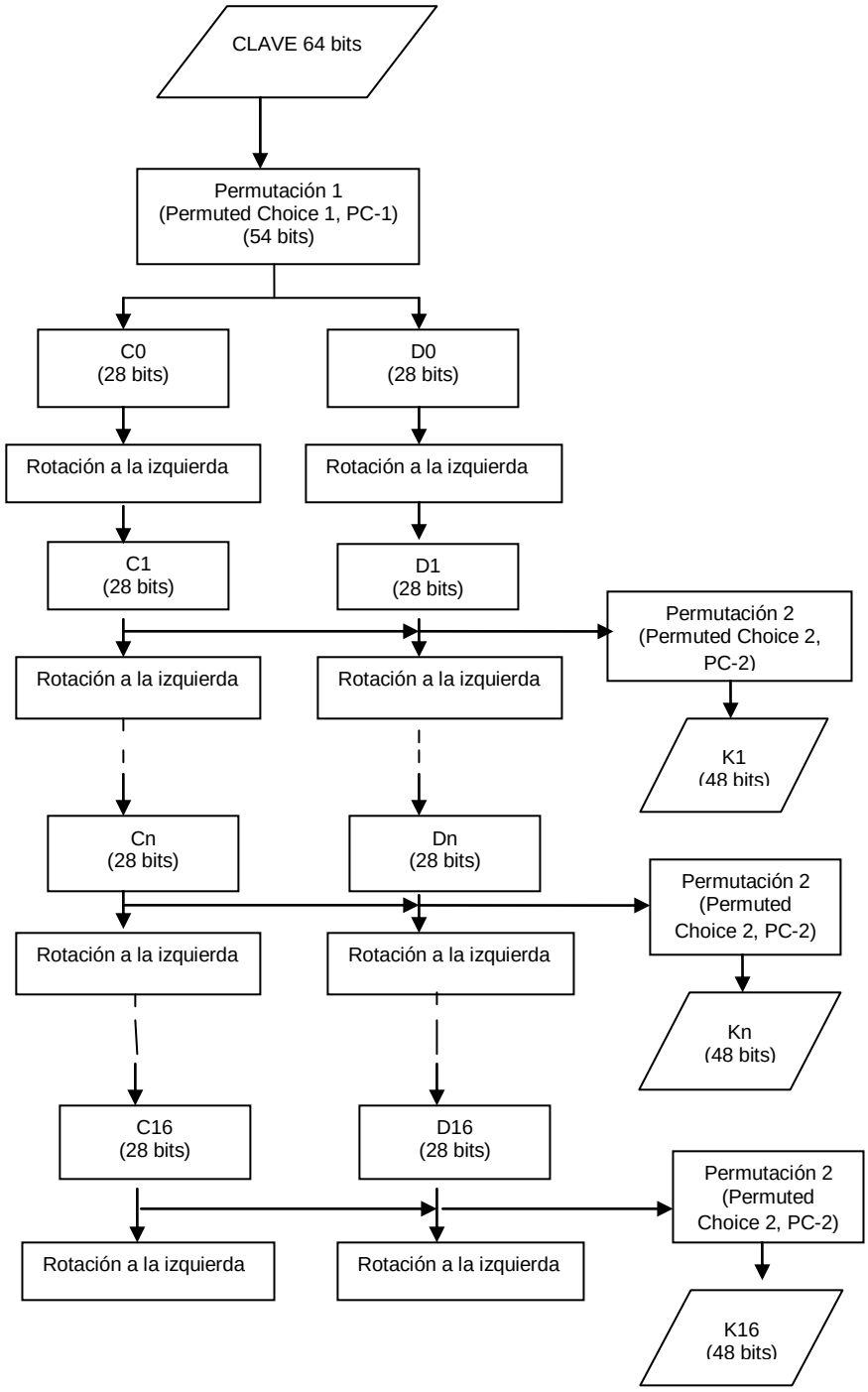


Figura 4.2 Calculo de las subclaves, K1

Se realiza una permutación inicial (PC-1) sobre la clave, y luego la clave obtenida se divide en dos mitades de 28 bits, cada una de las cuales se rota a izquierda un número de bits determinado que no siempre es el mismo. Ki se deriva de la elección permutada (PC-2) de 48 de los 56 bits de estas dos mitades rotadas.
La función f de la red de Feistel se compone de una permutación de expansión (E), que convierte el bloque correspondiente de 32 bits en uno de 48. Después realiza una or-

exclusiva con el valor Ki, también de 48 bits, aplica ocho S-Cajas de 6*4 bits, y efectúa una nueva permutación (P).

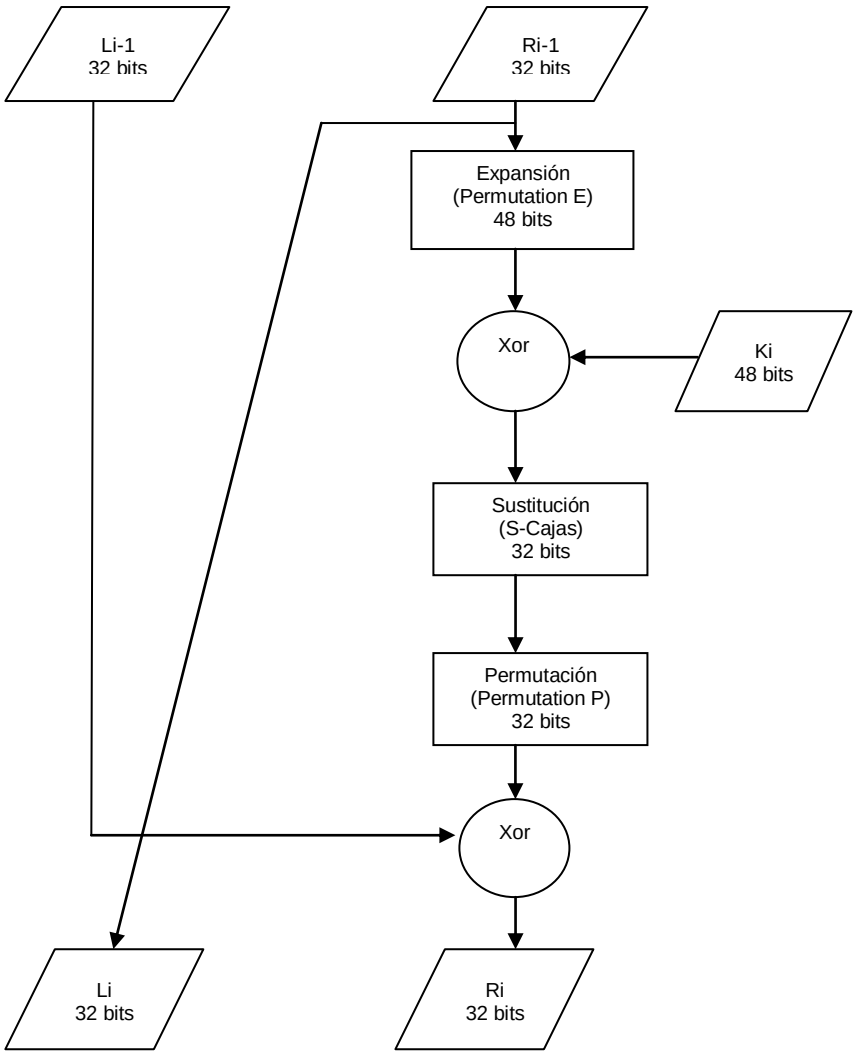


FIGURA 4.3 Ronda del Algoritmo DES

Para descifrar basta con usar el mismo algoritmo empleando las K_i en orden inverso.

4.2 ENCRIPTACIÓN

4.2.1 Algoritmo Descifrado

DES es un algoritmo de cifrado en bloques simétrico, el tamaño del bloque es de longitud fija de 64 bits, el algoritmo consta de dos permutaciones, una al inicio conocida como P1, la cual se muestra a continuación:

Tabla antes la Permutación

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
17	**18**	**19**	**20**	**21**	**22**	**23**	**24**
25	**26**	**27**	**28**	**29**	**30**	**31**	**32**
33	**34**	**35**	**36**	**37**	**38**	**39**	**40**
41	**42**	**43**	**44**	**45**	**46**	**47**	**48**
49	**50**	**51**	**52**	**53**	**54**	**55**	**56**
57	**58**	**59**	**60**	**61**	**62**	**63**	**64**

Se separan el mensaje original de 64 bits en bloques de 8 bits, previo a la permutación, la tabla *P1* muestra el resultado de la permutación.

Figura 4.4

Permutación Inicial (P1)

58	**50**	**42**	**34**	**26**	**18**	**10**	**2**
60	**52**	**44**	**36**	**28**	**20**	**12**	**4**
62	**54**	**46**	**38**	**30**	**22**	**14**	**6**
64	**56**	**48**	**40**	**32**	**24**	**16**	**8**
57	**49**	**41**	**33**	**25**	**17**	**9**	**1**
59	**51**	**43**	**35**	**27**	**19**	**11**	**3**
61	**53**	**45**	**37**	**29**	**21**	**13**	**5**
63	**55**	**47**	**39**	**31**	**23**	**15**	**7**

En la parte superior se puede observar que se encuentran los números pares.

En la parte inferior se puede observar que se encuentran los números impares.

Figura 4.5

Tabla formada para la Permutación Inicial P1

Permutación Inicial (P1)															
58	**50**	**42**	**34**	**26**	**18**	**10**	**2**	**60**	**52**	**44**	**36**	**28**	**20**	**12**	**4**
62	**54**	**46**	**38**	**30**	**22**	**14**	**6**	**64**	**56**	**48**	**40**	**32**	**24**	**16**	**8**
57	**49**	**41**	**33**	**25**	**17**	**9**	**1**	**59**	**51**	**43**	**35**	**27**	**19**	**11**	**3**
61	**53**	**45**	**37**	**29**	**21**	**13**	**5**	**63**	**55**	**47**	**39**	**31**	**23**	**15**	**7**

Después de recibir un bloque de entrada de 64 bits, el primer paso consiste en aplicar al bloque de entrada la permutación P1, teniendo como resultado un orden de salida que se identifica leyendo la tabla de izquierda a derecha y de arriba abajo. Significa que el bit del lugar 58 del mensaje de entrada, después de la permutación ocupara la posición 1 y así sucesivamente.

Una vez realizada la permutación, los 64 bits se dividen en sub-bloques Left y Rigth (L_i y R_i) de 32 bits cada uno. En estas condiciones, DES esta definido por las ecuaciones:

$$\mathbf{L_i = R_{i-1}}$$

$$\mathbf{R_i = L_{i-1} \oplus f\,(Ri_{-1}\,,\, K_i)}$$

El valor de i=16 représenta las 16 vueltas del algoritmo. A continuación se explicará con un ejemplo:

Ejemplo de permutación P1

Mensaje a cifrar = Denytamo

Decimal	Carácter	Binario
97	a	01100001
68	D	01000100
101	e	01100101
109	m	01101101
110	n	01101110
111	o	01101111
116	t	01110100
121	y	01111001

D	01000100
e	01100101
n	01101110
y	01111001
t	01110100
a	01100001
m	01101101
o	01101111

Utilizando la tabla de permutación P1 tenemos

Tabla antes la Permutación

0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	0	0	1
0	1	1	1	0	1	0	0
0	1	1	0	0	0	0	1
0	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1

Permutación Inicial (P1)

1	1	1	1	1	1	1	1
0	0	0	1	1	0	0	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0
1	1	0	0	1	1	0	0
1	0	0	0	0	1	0	0

Tenemos el resultado de la permutación P1, en la parte superior se muestran los bits que forman el sub-bloque Lo, y en la inferior los bits del sub-bloque Ro, dando como resultado:

$$L_0 = 11111111\ \ 00011000\ \ 11010111\ \ 11101010$$

$$R_0 = 00000000\ \ 11111110\ \ 11001100\ 10000100$$

Sub-bloques iniciales

4.2.2 Permutación E

La salida de Ro es de 32 bits, se utiliza la permutación E, con el propósito de expandir a 48 bits y así poder realizar la suma de la OR Exclusiva con al clave Ki.

Bits duplicados			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

Permutación E					
32	**1**	**2**	**3**	**4**	**5**
4	**5**	**6**	**7**	**8**	**9**
8	**9**	**10**	**11**	**12**	**13**
12	**13**	**14**	**15**	**16**	**17**
16	**17**	**18**	**19**	**20**	**21**
20	**21**	**22**	**23**	**24**	**25**
24	**25**	**26**	**27**	**28**	**29**
28	**29**	**30**	**31**	**32**	**1**

En las tablas de la figura 4.6, en la de la izquierda se muestran los bits duplicados para la expansión a 48 bits en la permutación E. Aquí se procede a la suma de la OR exclusiva.

Ejemplo de la permutación E

Al tener la secuencia de Ro de 32 bits, es necesario aplicar la permutación E

R_0 = 0000 0000 1111 1110 1100 1100 1000 0100

32 Bits	Permutación E
0 0 0 0	0 0 0 0 0 0
0 0 0 0	0 0 0 0 0 1
1 1 1 1	0 1 1 1 1 1
1 1 1 0	1 1 1 1 0 1
1 1 0 0	0 1 1 0 0 1
1 1 0 0	0 1 1 0 0 1
1 0 0 0	0 1 0 0 0 0
0 1 0 0	0 0 1 0 0 0

El resultado de la permutación E(Ro) es:

$E(R_0)$ = 000000 000001 011111 111101 011001 011001 010000 001000

4.2.3 Generación de la Subclave Ki

La clave Ki tiene un valor inicial de 64 bits y es fija.

Tabla de 64 bits

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Tabla de 56 bits

1	**2**	**3**	**4**	**5**	**6**	**7**
9	**10**	**11**	**12**	**13**	**14**	**15**
17	**18**	**19**	**20**	**21**	**22**	**23**
25	**26**	**27**	**28**	**29**	**30**	**31**
33	**34**	**35**	**36**	**37**	**38**	**39**
41	**42**	**43**	**44**	**45**	**46**	**47**
49	**50**	**51**	**52**	**53**	**54**	**55**
57	**58**	**59**	**60**	**61**	**62**	**63**

Se muestra la tabla que se utilizará para realizar la permutación PC1, donde se nota la falta de bits de paridad de cada byte. Esos no aportan ninguna información.

Figura 4.7

Permutación PC1

57	**49**	**41**	**33**	**25**	**17**	**9**
1	**58**	**50**	**42**	**34**	**26**	**18**
10	**2**	**59**	**51**	**43**	**35**	**27**
19	**11**	**3**	**60**	**52**	**44**	**36**
63	**55**	**47**	**39**	**31**	**23**	**15**
7	**62**	**54**	**46**	**38**	**30**	**22**
14	**6**	**61**	**53**	**45**	**37**	**29**
21	**13**	**5**	**28**	**20**	**12**	**4**

Esta se utiliza para realizar la permutación inicial en la generación de la sub clave Ki, para cada vuelta.

Figura 4.8

Una vez realizada la permutación los 56 bits se dividen en sub-bloques Ci y Di de de 28 bits. En estas condiciones la clave está definida por las ecuaciones:

$$C_i = LS\ (C_{i-1}) \quad D_i = LS(D_{i-1})$$

$$K_i = PC2\ (C_i\ ,\ D_i)$$

Ejemplo Permutación PC1

Clave K= Santiago

Decimal	Carácter	Binario
97	a	01100001
103	g	01100111
105	i	01101001
110	n	01101110
111	o	01101111
83	S	01010011
116	t	01110100

S 01010011
a 01100001
n 01101110
t 01110100
i 01101001
a 01100001
g 01100111
o 01101111

Utilizando la permutación PC1 obtenemos:

Tabla de 64 bits de K_i Inicial

0	1	0	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	0	1	0	0
0	1	1	0	1	0	0	1
0	1	1	0	0	0	0	1
0	1	1	0	0	1	1	1
0	1	1	0	1	1	1	1

Tabla de 56 bits

0	1	0	1	0	0	1
0	1	1	0	0	0	0
0	1	1	0	1	1	1
0	1	1	1	0	1	0
0	1	1	0	1	0	0
0	1	1	0	0	0	0
0	1	1	0	0	1	1
0	1	1	0	1	1	1

Permutación PC1

0	0	0	0	0	0	0
0	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	0	0	0	0	0
1	1	0	0	0	1	0
1	1	1	0	0	1	1
0	0	1	0	0	1	0
1	0	0	1	0	0	1

Sub-bloque C_0 = 0000000 0111111 1111111 1100000

Sub-bloque D_0 = 1100010 1110011 0010010 1001001

Sub-bloques iniciales

4.2.4 Desplazamiento LS(…)

Este se aplica a los sub-bloques de longitud fija de 7 bits (Ci y Di), donde LS es un desplazamiento circular a la izquierda de 1 o 2 bits del entero binario que toma el argumento de acuerdo a la siguiente tabla:

Vuelta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
No.Bits desplazados. Izda.	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

En la figura 4.9 tiene el propósito de comprender mejor el proceso.

Ejemplo de Permutación LS(…)

Al tener los sub-bloques Co y Do el siguiente paso es el desplazamiento LS, se marcan los 2 primeros bits con el propósito de identificar el resultado del desplazamiento.

Sub-bloque C_0 = 0000000 0111111 1111111 1100000

Sub-bloque D_0 = 1100010 1110011 0010010 1001001

al ser la primer vuelta, el desplazamiento es de un bit a la izquierda como se indica en la tabla , dando como resultado C1 y D1.

Sub-bloque C_1 = 0000000 1111111 1111111 1000000

Sub-bloque D_1 = 1000101 1100110 0100101 0010011

Sub-bloques después del desplazamiento LS(..)

4.2.5 Permutación PC2

Esta se conoce como permutación de comprensión, dada por las operaciones de concatenar y permutar C_i y D_i, se van a comprimir de 56 a 48 bits para obtener la clave K_i, posteriormente será utilizada en la función ($f(R_{i-1}, K_i)$). El orden de concatenar Ci y Di, es utilizando primero los 28 bits de Ci y posteriormente los 28 bits de Di, la tabla de PC2 es una tabla d 8x6, dando como resultado 8 bloques de 6 bits.

Tabla (Ci, Di)

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

(a)

tabla 8X6 bits

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48

(b)

En las tablas de la figura 4.10 . Se muestra el orden para generar la tabla de permutación PC2, y en la tabla (a) se marcan los bits eliminados para comprimir a 48.

Permutación PC2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Figura 4.11 Permutación PC2 final

Ejemplo permutación PC2

Concatenando C_i, D_i

0000000 1111111 1111111 1000000 1000101 1100110 0100101 0010011

Los 56 bits de entrada en la permutación PC2

Tabla (C_i, D_i)

0 0 0 0 0 0 0

1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 0 0 0 0 0 0

1 0 0 0 1 0 1

1 1 0 0 1 1 0

0 1 0 0 1 0 1

0 0 1 0 0 1 1

Permutación PC2

1 1 1 0 0 0

0 0 1 0 1 1

0 1 1 0 0 1

1 0 0 1 1 0

1 1 0 1 1 1

0 1 0 0 1 0

1 1 0 1 0 0

0 0 0 1 1 0

El resultado de haber realizado la permutación PC2 es la generación de la clave K_1 siendo:

K_1 = PC2 (C_1, D_1) =

111000 001011 011001 100110 110111 010010 110100 000110

Clave K_1

Las operaciones *LS*(..) y PC2, se repiten 15 veces para así obtener las 15 sub claves de cifrado restantes.

4.2.6 Función $f(R_{i-1}, K_i)$

Al tener la clave K_i y la expansión de (R_0), el siguiente paso es la función $f(R_{i-1}, K_i)$, la cual consta de tres procesos (Suma OR exclusivo, ocho funciones no lineales, Permutación P), siendo las ocho funciones lineales la mayor virtud del algoritmo, se han propuesto modificaciones en varios procesos, pero cualquier modificación realizada, en las funciones lineales hace débil al algoritmo

SUMA OR EXCLUSIVA

Con la clave K_i y $E(R_{i-1})$, se procede a realizar la suma OR exclusiva, al realizar la operación se tiene un 50% de certeza que el siguiente bit sea 1, con lo cual aumenta la dificultad de poder descifrar el mensaje. La operación OR exclusiva se ejemplifica en la siguiente tabla:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 4.12 OR Exclusiva

Se encuentra definido por la ecuación:

$E(R_{i-1}) \oplus K_i$

La clave K_i y R_{i-1} se encuentra formado por 8 bloques de 6 bits cada, con lo cual es posible realizar la suma OR exclusiva.

Ejemplo suma OR exclusiva

Retomando los bits de la expansión del sub-bloque R_0 y la sub clave K_1, se procede a realizar la suma OR exclusiva.

$E(R_0)$ = 000000 000001 011111 111101 011001 011001 010000 001000

⊕

K_1 = 111000 001011 011001 100110 110111 010010 110100 000110

111000 001010 000110 011011 101110 001011 100100 001110

Resultado de la suma OR exclusiva $E(R_0) \oplus K_1$

FUNCIONES NO LINEALES

La cadena de bits obtenida de la suma OR exclusiva, se subdivide en 8 bloques de 6 bits, como se puede apreciar en el ejemplo anterior, siendo cada bloque ($b_6b_5b_4b_3b_2b_1$) la entrada a una de las funciones no lineales, conocidas como *cajas*, el resultado de la operación es un número con valor entre 0 a 15, representado con cuatro bits, concatenados forman una cadena de 32 bits, la cual será la entrada para la permutación P. La posición de los bits dentro de cada caja se encuentra definida por la fila b_6b_1 y la columna $b_5b_4b_3b_2$ de la caja, el orden de los valores numéricos de las cajas se puede ver en la siguiente imagen:

Ejemplo función no lineal (cajas)

$E(R_0) \oplus K_1$ = 111000 001010 000110 011011 101110 001011 100100 001110

En la Caja S_1 se utiliza el primer bloque **111000**, en la Caja S_2 se utiliza el segundo bloque y así sucesivamente.

b_6b_1	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	**1100**	1101	1110	1111	
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
b_6b_1	0000	0001	0010	0011	0100	**0101**	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
b_6b_1	0000	0001	0010	**0011**	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
00	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
01	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
10	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
11	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
b_6b_1	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	**1101**	1110	1111	
00	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
01	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
10	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
11	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	

b_6b_1	0000	0001	0010	0011	0100	0101	0110	**0111**	1000	1001	1010	1011	1100	1101	1110	1111	
00	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
01	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
10	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
11	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
b_6b_1	0000	0001	0010	0011	0100	**0101**	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
00	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
01	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
10	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
11	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
b_6b_1	0000	0001	**0010**	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
00	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
01	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
10	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
11	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
b_6b_1	0000	0001	0010	0011	0100	0101	0110	**0111**	1000	1001	1010	1011	1100	1101	1110	1111	
00	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
01	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
10	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
11	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Figura 4.13 Cajas

La salida del bloque 1 es: 3 (0011), el procedimiento se repite en las siguientes 7 cajas obteniendo como resultado:

$$\mathbf{E(R_0) \oplus K_1 = \ 0011\ \ 1011\ \ 1110\ \ 1010\ \ 1000\ \ 1100\ \ 1011\ \ 0001}$$

$$\mathbf{3\quad 11\quad 14\quad\quad 10\quad 8\quad 12\quad 11\quad 1}$$

Resultado de las funciones no lineales

PERMUTACION P

El último paso de la función $f(R_{i-1}, K_i)$, es una permutación P, cuyo resultado se sumará con la salida del sub-bloque L_i, dando origen a la entrada del sub-bloque R_i

Tabla

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

Permutación P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Figura 4.13 Tabla para la permutación P

Ejemplo de permutación P

Retomando el ejemplo se realiza el último paso de la $f(R_{i-1}, K_i)$, recordando que dicho proceso se repite 15 veces.

$$\mathbf{E(R_0) \oplus K_1 = 0011\ \ 1011\ \ 1110\ \ 1010\ \ 1000\ \ 1100\ \ 1011\ \ 0001}$$

Resultado de las funciones no lineales

Tabla

0	0	1	1
1	0	1	1
1	1	1	0
1	0	1	0
1	0	0	0
1	1	0	0
1	0	1	1
0	0	0	1

Obteniendo como resultado de la permutación:

$f(R_{i-1}, K_i)$ = **0101 0011 0100 1001 0100 1111 0100 1111**

Permutación P

0	1	0	1
0	0	1	1
0	1	0	0
1	0	0	1
0	1	0	0
1	1	1	1
0	1	0	0
1	1	1	1

4.2.7 Suma Li $\oplus$ Ri

El registro de bits obtenido por la función $f(R_{i-1}, K_i)$, es sumado con un OR exclusivo con el registro de bits de L_0, dando como resultado la entrada para el siguiente sub-bloque de R_i. El registro de bits realizado en la permutación inicial P1, que originó al sub-bloque R_0, es la entrada para el siguiente sub-bloque L_i, el proceso se repite 15 veces, siendo el último proceso la permutación $P1^{-1}$.

Ejemplo Suma $L_i \oplus f(R_{i-1}, K_i)$

$f(R_{i-1}, K_i)$ = 0101 0011 0100 1001 0100 1111 0100 1111

$\oplus$

L_0 = 1111 1111 0001 1000 1101 0111 1110 1010

1010 1100 0101 0001 1001 1000 1010 0101

Resultado de la suma OR exclusiva $L_0 \oplus f(R_0, K_1)$

Obteniendo los registros de 32 bits para el siguiente sub-bloque L_i y R_i como se muestra a continuación:

L_1 = **0000 0000 1111 1110 1100 1100 1000 0100**

R_1 = **1010 1100 0101 0001 1001 1000 1010 0101**

Donde se verifican las ecuaciones:

$$L_1 = R_0$$

00000000111111101100110010000100 =
00000000111111101100110010000100

$$R_1 = L_0 \oplus f(R_0, K_1)$$

10101100010101011001100010100101 =

11111111000110001101011111101010$\oplus$01010011010011010100111101001111

4.2.8 Permutación $P1^{-1}$

La permutación inversa se define por la siguiente tabla, siendo la salida, el cifrado del mensaje.

Tabla antes la Permutación

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Permutación Final ($P1^{-1}$)

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	56	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figura 4.15 Permutación $P1^{-1}$

Ejemplo de Permutación $P1^{-1}$

Si tomamos los valores que tenemos de L_1 y R_1, suponiendo los valores de L_{16} y R_{16}, podemos realizar el procedimiento, como muestra del resultado final, cabe mencionar si el mensaje es mayor de 64 bits, se utilizan los bloques necesarios para dividir el mensaje original, si un bloque formado por un mensaje, no tiene la longitud de 64 bits, se rellena utilizando 0.

L_{16} = **10101100 01010001 10011000 10100101**

R_{16} = **00000000 11111110 11001100 10000100**

Concatenando L_{16} , R_{16}

10101100 01010001 10011000 10100101 00000000 11111110 11001100 10000100

(a)

1	0	1	0	1	1	0	0
0	1	0	1	0	0	0	1
1	0	0	1	1	0	0	0
1	0	1	0	0	1	0	1
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0
1	1	0	0	1	1	0	0
1	0	0	0	0	1	0	0

(b)

0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0
0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0
0	1	1	0	1	1	1	1

Tabla 4.14 Figura (a) antes de la Permutación (b) $P1^{-1}$

El cifrado del mensaje es: ◄(spacio){l4a8o

00010001 00100000 01101011 01101100 00110100 01100001 00111000 01101111

Decimal	Carácter	Binario
17	◄	**00010001**
32	(space)	**00100000**
123	{	**01111011**
108	L	**01101100**
52	4	**00110100**
97	A	**01100001**
56	8	**00111000**
111	O	**01101111**

Tabla 4.15

4.3 DESENCRIPTACIÓN O DESCIFRADO

El algoritmo se utiliza para obtener el mensaje original, con un sentido inverso al inicial, esto es, empezando con la entrada del mensaje cifrado de 64 bits, aplicando la permutación P1, dividir el mensaje en dos sub-bloques L_{16} y R_{16}, teniendo la subclaves calculadas previamente, se realiza el procedimiento descrito anteriormente la $f(R_{i-1}, K_i)$, realizando las 16 vueltas hasta obtener el mensaje en claro.

Ejemplo Permutación P1 Descifrado
Mensaje a Descifrar = ◄(spacio){l4a8o

Decimal	Carácter	Binario
17	◄	**00010001**
32	(space)	**00100000**
123	{	**01111011**
108	L	**01101100**
52	4	**00110100**
97	A	**01100001**
56	8	**00111000**
111	O	**01101111**

Tabla antes la Permutación

0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0
0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	0
0	0	1	1	0	1	0	0
0	1	1	0	0	0	0	1
0	0	1	1	1	0	0	0
0	1	1	0	1	1	1	1

Permutación Inicial (P1)

1	0	1	0	1	1	0	0
0	1	0	1	0	0	0	1
1	0	0	1	1	0	0	0
1	0	1	0	0	1	0	1
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0
1	1	0	0	1	1	0	0
1	0	0	0	0	1	0	0

Tablas 4.16 antes y después de la Permutación

L_{16} = 10101100 01010001 10011000 10100101

R_{16} = 00000000 11111110 11001100 10000100

Ejemplo Permutación E Descifrado

Al tener la secuencia de R_{16} de 32 bits, es necesario aplicar la permutación E, la cual se muestra a continuación.

R_{16} = 00000000 11111110 11001100 10000100

32 Bits	Permutación E
0 0 0 0	0 0 0 0 0 0
0 0 0 0	0 0 0 0 0 1
1 1 1 1	0 1 1 1 1 1
1 1 1 0	1 1 1 1 0 1
1 1 0 0	0 1 1 0 0 1
1 1 0 0	0 1 1 0 0 1
1 0 0 0	0 1 0 0 0 0
0 1 0 0	0 0 1 0 0 0

$E(R_{16})$ = 000000 000001 011111 111101 011001 011001 010000 001000

Ejemplo Suma OR exclusiva Descifrado

La suma OR exclusiva se realiza utilizando la ultima clave generada (K_{16}), en nuestro ejemplo utilizaremos la clave K_1, como la clave K_{16}.

$E(R_{16})$ = 000000 000001 011111 111101 011001 011001 010000 001000

$\oplus$

K_{16} = 111000 001011 011001 100110 110111 010010 110100 000110

111000 001010 000110 011011 101110 001011 100100 001110

Resultado de la suma OR exclusiva $E(R_{16}) \oplus K_{16}$

Ejemplo función no lineal (cajas) Descifrado

$E(R_0) \oplus K_1$ = 111000 001010 000110 011011 101110 001011 100100 001110

Como resultados de las operaciones no lineales (cajas) tenemos los siguientes:

$E(R_0) \oplus K_1$ =	**0011**	**1011**	**1110**	**1010**	**1000**	**1100**	**1011**	**0001**
	3	**11**	**14**	**10**	**8**	**12**	**11**	**1**

Resultado de las funciones no lineales

Ejemplo Permutación P Descifrado

Retomando el ejemplo se realiza el último paso de la $f(R_{i-1}, K_i)$, recordando que dicho proceso se repite 15 veces mas.

$E(R_0) \oplus K_1$ = 0011 1011 1110 1010 1000 1100 1011 0001

Tabla
0 0 1 1
1 0 1 1
1 1 1 0
1 0 1 0
1 0 0 0
1 1 0 0
1 0 1 1
0 0 0 1

Obteniendo como resultado de la permutación la siguiente salida:

$f(R_{i-1}, K_i)$ = 0101 0011 0100 1001 0100 1111 0100 1111

Permutación P
0 1 0 1
0 0 1 1
0 1 0 0
1 0 0 1
0 1 0 0
1 1 1 1
0 1 0 0
1 1 1 1

Ejemplo Suma $L_i \oplus f(R_{i-1}, K_i)$ Descifrado

$f(R_{i-1}, K_i)$ = 0101 0011 0100 1001 0100 1111 0100 1111

$\oplus$

L_{16} = 1010 1100 0101 0001 1001 1000 1010 0101

1111 1111 0001 1000 1101 0111 1110 1010

Resultado de la suma OR exclusiva $L_0 \oplus f(R_{i-1}, K_i)$

Obteniendo los registros de 32 bits para el siguiente sub-bloque L_i y Ri como se muestra a continuación:

L_{15} = **0000 0000 1111 1110 1100 1100 1000 0100**

R_{15} = **1111 1111 0001 1000 1101 0111 1110 1010**

Donde se verifican las ecuaciones:

$$L_{15} = R_{16}$$

00000000111111101100110010000100 =
00000000111111101100110010000100

$$R_1 = L_{16} \oplus f(R_{16}, K_{16})$$

11111111000110001101011111101010 =

10101100010100011001100010100101⊕01010011010010010100111101001111

Ejemplo Permutación $P1^{-1}$ Descifrado

Si tomamos los valores que tenemos de L_1 y R_1, suponiendo los valores de L_{16} y R_{16}, podemos realizar el procedimiento, como muestra del resultado final, cabe mencionar que el mensaje es mayor de 64 bits, se utilizan los bloques necesarios para dividir el mensaje original, si un bloque formado por un mensaje y este no tiene la longitud de 64 bits, se rellena utilizando 0.

L_1 = **1111 1111 0001 1000 1101 0111 1110 1010**

R_1 = **0000 0000 1111 1110 1100 1100 1000 0100**

Concatenando L_1 y R_1, da como resultado:

11111111 00011000 11010111 11101010 00000000 11111110 11001100 10000100

(a)

1	1	1	1	1	1	1	1	D	01000100
0	0	0	1	1	0	0	0	e	01100101
1	1	0	1	0	1	1	1	n	01101110
1	1	1	0	1	0	1	0	y	01111001
0	0	0	0	0	0	0	0	t	01110100
1	1	1	1	1	1	1	0	a	01100001
1	1	0	0	1	1	0	0	m	01101101
1	0	0	0	0	1	0	0	o	01101111

(b)

0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	0	0	1
0	1	1	1	0	1	0	0
0	1	1	0	0	0	0	1
0	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1

Tabla 4.17 Figura (a) antes de la permutación y (b) $P1^{-1}$

CAPITULO 5. IMPLEMENTACIÓN DEL ALGORITMO 'DES'

Este capítulo llega a la finalización de nuestro trabajo realizado, en el cual explicaremos brevemente ciertos códigos realizados en un lenguaje de alto nivel, utilizando la interface grafica de Visual Basic.NET.

Dentro de Visual con la denominación genérica ADO.NET se hace referencia a todos los servicios de acceso a datos disponibles en la plataforma .NET. Los elementos de ADO.NET están pensados para simplificar el acceso a los datos, sin perder por ello flexibilidad y eficacia.

ADO, gracias a la existencia de diversos controladores OLE DB, ofrece acceso a orígenes de datos de todo tipo y facilita al desarrollador las operaciones más habituales. Acceder a un cierto origen de datos mediante el proveedor OleDb y un controlador OLE DB supone, como es fácil deducir, una capa mas de código, lo cual implica mayor uso de recursos y menor rendimiento. El proveedor que en este caso utilizaremos de OLE DB se llama Microsoft OLE DB Provider for Microsoft Jet y su identificador es: Microsoft.Jet.OLEDB.4.0.

Este controlador sirve para abrir archivos de Excel y Access.

ConnOleDb.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0…..

Este es un ejemplo de la conexión con Microsoft Excel

En Visual Basic.NET que conecte con la base de datos Excel.

'Necesitamos las clases del proveedor OLE DB

```
Imports System.Data.OleDb

Module Module1

    Sub Main()

`Creamos el objeto Connection
        Dim ConnOleDb As New OleDbConnection()

`y establecemos la cadena de conexión
        ConnOleDb.ConnectionString = _
        "Provider =Microsoft.Jet.OLEDB.4.0; " & _
        "Data Source=\C:User\vero\Documents\OLIMPIADA.xls; " & _
        "Extended Properties ='Excel 12.0; HDR=Yes'"
`el Extended Properties sirve para enviar al controlador parametros
exclusivos, indicando lo que va abrirse en uan hoja de excel.
```

```
        Try     `intentamos abrir la conexion
            ConnOleDb.Open()

`si esta abierta
            If ConnOleDb.State = ConnectionState.Open Then

`hemos tenido exito, asi que mostrara unos datos
                Console.WriteLine("se ha establecido la conexion")
                Console.WriteLine("ConnectionString='"&
ConnOleDb.ConnectionString & "'")
                Console.WriteLine("ConnectionTimeout="&
ConnOleDb.ConnectionTimeout)
                Console.WriteLine("Database=" & ConnOleDb.Database)

                ConnOleDb.Close()      `y la cerramos

                Else                   `si no esta abierta, lo indicamos
                Console.WriteLine("La conexion no esta abierta")

            End If
        Catch ex As Exception `si se produce un error lo indicamos
            Console.WriteLine("*** Error al intentar la conexión***")


        End Try

    End Sub

End Module
```

Este es el pequeño código que nos muestra la conexión de un archivo por medio de OLEDB para poder acceder a los datos de una base en Microsoft excel.

Aquí procedemos a elaborar la rutina que nos muestra nuestro algoritmo "DES".

Creamos una interfaz grafica en Visual Basic

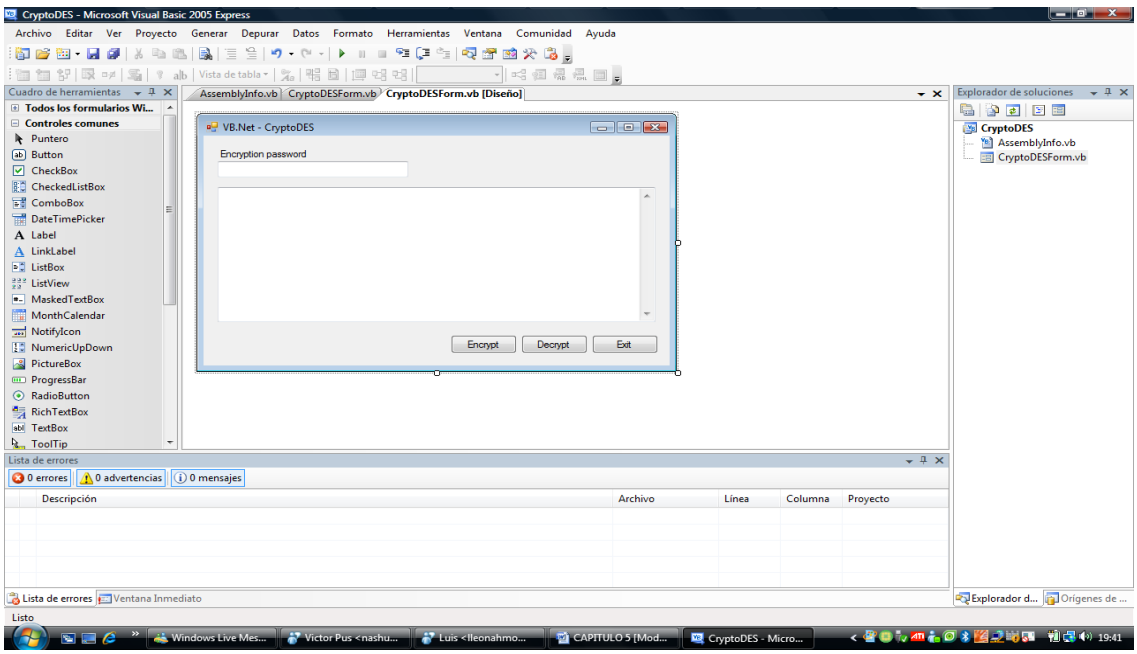


Figura 5.1 Interfaz grafica

El primer paso es agregar la referencia de la librería que utilizaremos, nos vamos a proyecto, agregar referencia y la buscamos; CryptoCOM 1.0Library.

Agregada la referencia damos doble clic sobre nuestro botón de Encrypt y nos abre una ventana con el código y a continuación agregaremos las llamadas para el algoritmo.

```
Private Sub encrypt_Click()

'Aquí creamos el objeto de encriptación'

   Dim des As New CryptoCOMLib.CryptoTripleDES

   Dim b64 As New CryptoCOMLib.CryptoBase64

'generamos y asignamos la llave de encriptación'

Call  des.DeriveKeyFromPassword(pwd.Text,Empty,1234)

'Llamamos al método para encriptar el texto'

Call des.EncryptText(textToEncrypt.Text)

textToEncrypt.Text = b64.encrypt(des.Result)

End Sub
```

El código obtenido de la encriptación es en binario, y el criptoBase64 lo convierte a código ASCII para poderlo mostrar en la pantalla.

Ahora salimos y damos doble clic sobre el botón de Decrypt y nos abre de nuevo la ventana del código donde agregaremos lo siguiente:

```
Private Sub Decrypt_Click()

'Aquí creamos el objeto a desencriptar'

    Dim des As New CryptoCOMLib.CryptoTripleDES

    Dim b64 As New CryptoComLib.CryptoBase64

Call b64.decrypt (textToEncrypt.Text)

Call des.DeriveKeyFromPAssword(pwd.Text,Empty,1234)

textToEncrypt.Text  = des.DecryptText(b64.Result)

End Sub.
```

A continuación mostrare el ejemplo de como se encripta y desencripta un texto, con el código del algoritmo.

En la figura 5.2 se introduce el password y un pequeño texto.

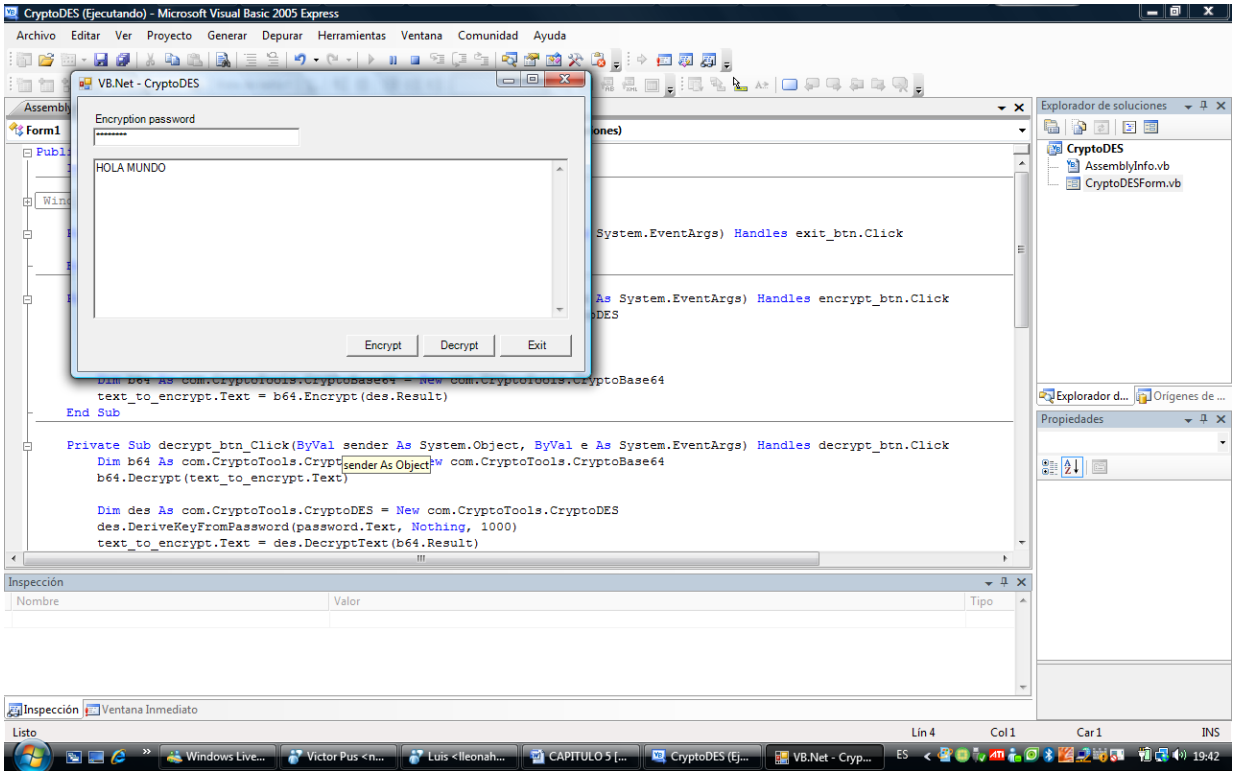


Figura 5.2 Introducción de password y texto

En la figura 5.3 se va a encriptar el texto.

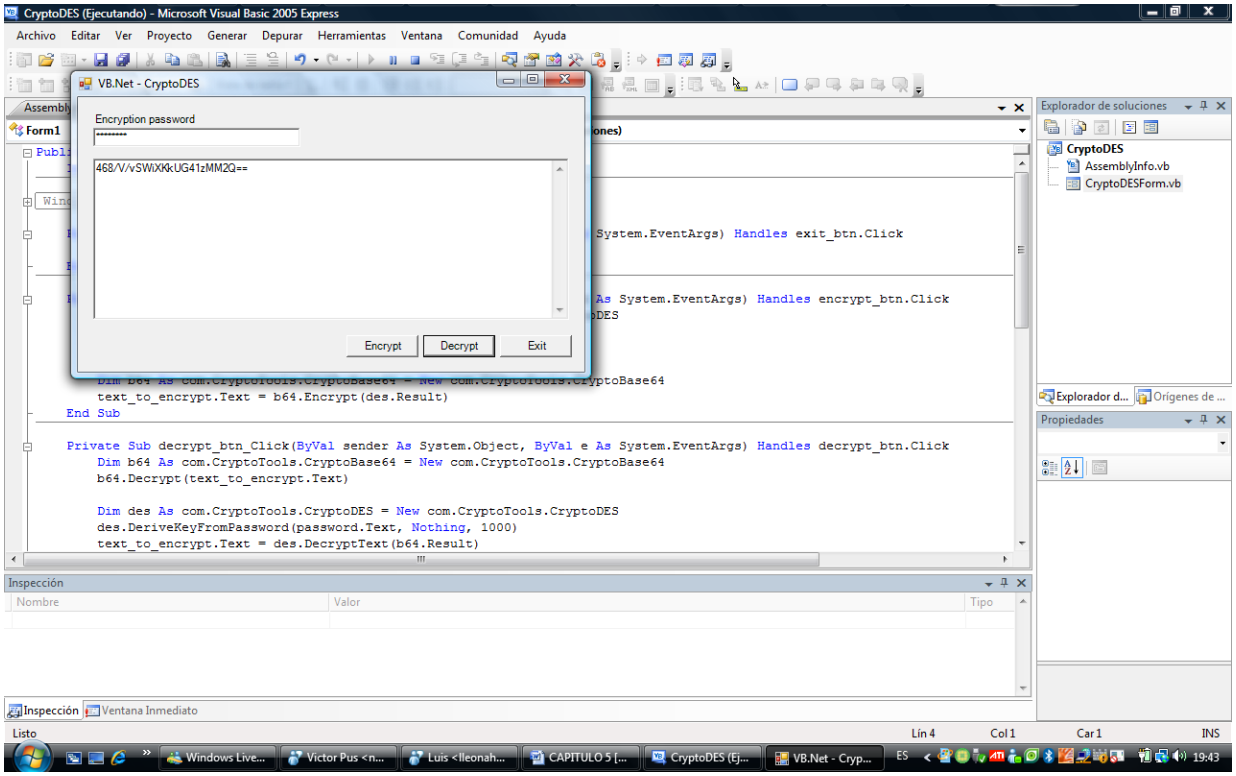


Figura 5.3 Encriptación

En la figura 5.4 se desencripta el texto.

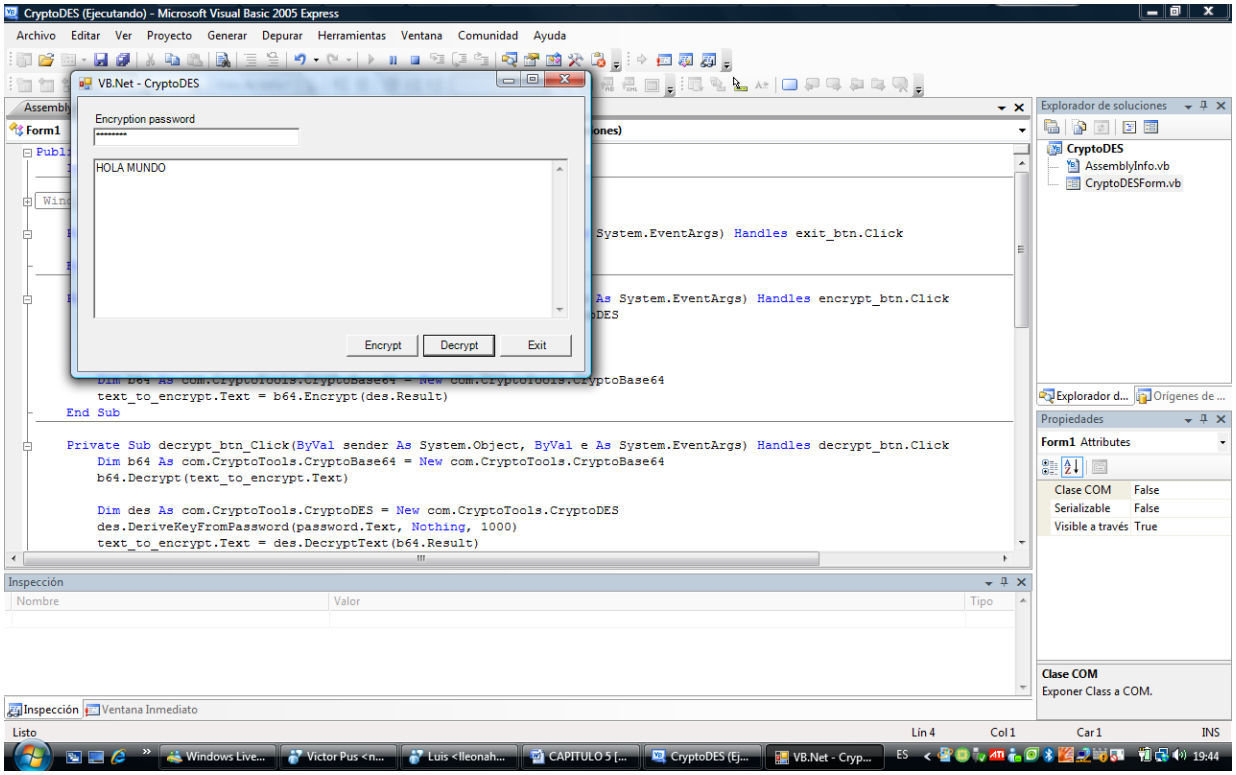


Figura 5.4 Desencriptación

El primer código forma parte de nuestro proyecto ya que con este podemos abrir los archivos en Excel o Access y de este modo comenzar el proceso de encriptación, por el cual el segundo código nos muestra como se realizaría el proceso para poder encriptar y desencriptar utilizando nuestro algoritmo "DES". Los dos realizados en nuestro lenguaje de lato nivel como es Visual Basic.NET

CONCLUSIONES

La seguridad informática hoy en día ha cobrado una mayor fuerza, además de un avance significativo, las tecnologías de seguridad se han vuelto más robustas y menos complicadas que al principio cuando se hablaba de obtener seguridad, pero sin lugar a duda se puede tener una seguridad que sea integral con base a un buen análisis.

Pero cabe mencionar que como se sabe la seguridad es relativa y dinámica, por lo cual se puede decir que es todo un proceso continuo en donde lo que ahora es seguro, mañana seguramente ya no lo será, no se puede confiar absolutamente en que lo implementado nos dará por siempre la seguridad por que en ese preciso momento no ha pasado nada, o seguramente no pasará nada esta es una utopía de que no se requiere revisar continuamente los lineamientos de seguridad. Sin una estrategia integral y sin ser aislada, no se tendrá una visión de lo que realmente se requiere, la seguridad no se arregla si sólo se le invierte en tecnología ya que el factor humano es la base para resolver y solventar la seguridad en donde se requiera ser implementada.

Para la realización de este tema me llevó a encontrar toda una serie de información, lo importante aquí fue documentar la apropiada para sustentar este trabajo, por lo cual muchos de los conceptos teóricos debían estar bien fundamentados con el fin de aportar los conocimientos para la realización propia del tema.

En momentos se presentaban nuevas tecnologías o nuevas amenazas que tenían que ser replanteadas debido al dinamismo que presenta la seguridad, uno de los problemas que fueron de los mas difíciles de solventar y que aun en día presenta dificultad es el llevar la concientización de el rol de cada uno de los miembros de la organización, por la cual el llevar acabo un plan de tal magnitud seguirá llevando tiempo, pero en este punto se llego a la conclusión de que una concientización y entrenamiento adecuado equivale más del 50% del éxito al implementar las medidas u otros aspectos en la seguridad, esto es debido a que la gente es el factor mas vulnerable en todo este entorno

A continuación se presenta un resumen por capítulo concluyendo los puntos más importantes que se obtuvieron.

CAPITULO 1. BASES DE DATOS

En este capitulo comenzamos viendo los antecedentes de las bases de datos, desde las cintas magnéticas hasta el surgimiento de la World Wide Web, y ahora en nuestros tiempos con los dispositivos de almacenamiento como las memorias con gran capacidad entre otras. Aquí mencionamos que las bases de datos son un enorme receptáculo el cual guarda información. Mencionamos los objetivos de las bases de datos así como sus ventajas y desventajas, la visión que tienes estas, el tipo de modelos de los datos, sus lenguajes, su arquitectura, los tipos de administradores para la misma, la estructura de un sistema de bases de datos y los importantes sistemas de gestión de las bases de datos. Dimos un enfoque general de lo que es una base de datos y su funcionamiento.

CAPITULO 2. TERMINOS GENERALES DE SEGURIDAD INFORMATICA

En este capitulo iniciamos con los antecedentes sobre la seguridad informática, dando así una definición de la seguridad y conociendo como han cambiado con el paso del tiempo, por lo tanto nos menciona que es lo que se requiere implementar, además de saber los servicios de seguridad, criterios, modelos, y estándares. El tipo de amenazas que existen y como van cambiando con el tiempo dando pauta así a las vulnerabilidades y amenazas que surgen para nuestra base de datos, y tratando de controlar con algunas contramedidas establecidas. Manejando también la confidencialidad, disponibilidad e integridad de los datos. Básicamente tratamos de dar un enfoque sobre la seguridad y posibles soluciones para la misma.

CAPITULO 3. CRIPTOGRAFIA

Se define a la criptografía como el cifrado y descifrado de mensajes, el cual es la transformación de un mensaje claro a un mensaje mediante el uso de claves. Checamos sus antecedentes, conceptos básicos y como se maneja este tipo de claves en un sistema de seguridad en la informática, dando así a conocer algunos sistemas de cifrado, aquí es donde conocemos nuestro algoritmo descifrado para ponerlo en práctica. El algoritmo “DES” y sus modos que nos sirven para asegurar nuestra información en las base de datos.

CAPITULO 4. METODOLOGIA DEL ALGORITMO "DES"

Este es el capitulo mas importante del proyecto ya que nos muestra como de desarrolla nuestro algoritmo descifrado. Como el nombre del capitulo lo dice es la metodología de nuestro algoritmo, el cual se basa en operaciones lógicas booleanas, consiste en la reducción de la longitud de clave y de sus bloques, DES cifra bloques de 64 bits, mediante permutación y sustitución, usando una clave de 64 bits de los cuales 8 son de paridad, produciendo así 64 bits cifrados. En pocas palabras convierte los datos visibles en datos no visibles para el usuario, dando claves para no entender el contenido del mismo. A esto se le llama encriptación de datos pero como en todo también nos explica el mecanismo de desencriptación de los mismos para volverlo visible. Es una forma segura de manejo de información para las empresas que en la actualidad se preocupan por males manejos de información.

CAPITULO 5. IMPLEMENTACIÓN DEL ALGORITMO "DES"

Aquí concluimos nuestro proyecto realizando unas pruebas con nuestro código aplicado en un lenguaje de alto nivel como lo es el Visual Basic, creamos un ejemplo del algoritmo con resultados obtenidos satisfactoriamente, y también utilizamos un ejemplo de cómo abrir una base de datos para su uso. Cabe mencionar que al crear nuestra interfaz grafica en Visual este compilador tiene una librería llamada Crypto Tools en la cual nos vamos a sus referencias y existe la del Algoritmo DES que nos hace más fácil el trabajo ya que al correrlo este hace el proceso del algoritmo obteniendo así los resultados previstos.

Finalmente se puede establecer que el objetivo fundamental de este proyecto era proporcionar una herramienta de seguridad para los datos. Empleando una metodología de encriptación que en este caso es nuestro algoritmo "DES".

Empleando todo tipo de herramientas, metodologías y técnicas para poder proteger la información en cualquier Base de Datos, fortaleciendo así la seguridad de cualquier empresa para el mejor manejo de los mismos. Que esto lleva tener mayor seguridad informática a nivel empresarial, solucionando uno de los importantes problemas a nivel mundial.

La enseñanza que obtuve de manera personal es que la carrera de computación tiene un campo muy amplio y requiere mayor seguridad en el manejo de los datos, actualizándose constantemente con al seguridad en informática. Y básicamente tratar de concienciar a la gente de la importancia del manejo de los datos y la problemática que tiene esto, además de brindarles soluciones para un mejor funcionamiento empresarial.

GLOSARIO

Algoritmo: es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien lo ejecute. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

Encriptación: es el proceso para volver ilegible información considera importante. La información una vez encriptada sólo puede leerse aplicándole una clave.

Plataforma: En informática, determinado software y/o hardware con el cual una aplicación es compatible y permite ejecutarla.

Proliferación: multiplicarse abundantemente el número o la cantidad de una cosa.

Automatización: El termino Automatización viene -como muchos de ustedes lo supondrán- de la palabra griega "auto" y significa la ejecución por medios propios de un proceso, en el que materia, información o energía es cambiado o transformado.

DB2: Universal Database (Base de datos Universal), en base a 2.

Oracle: Oracle es básicamente un herramienta cliente/servidor para la gestión de base de datos, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general.

Ingres: sistema de gestión de Bases de Datos.

SQL: (Structured Query Language) Lenguaje de consulta estructurado. SQL es un lenguaje formal declarativo, estandarizado ISO, para manipular información en una base de datos.

Word Wide Web: En informática, la **World Wide Web**, cuya traducción podría ser *Red Global Mundial* o "Red de Amplitud Mundial", es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

Interface: El interfaz, en informática, es un elemento de conexión que facilita el intercambio de datos, como por ejemplo el teclado, un tipo de interfaz entre el usuario y la computadora.

Atomicidad: se dice que una operación es atómica cuando es imposible para otra parte de un sistema encontrar pasos intermedios. Si esta operación consiste en una serie de pasos, todos ellos ocurren o ninguno. Por ejemplo en el caso de una transacción bancaria o se ejecuta tanto el depósito y la deducción o ninguna acción es realizada. Es una característica de los sistemas transaccionales. El concepto también es relevante cuando se programa con hilos de ejecución.

Items: Se usa para hacer distribución de artículos o capítulos en una escritura y también como señal de adición.

Utópica: Sistema o proyecto, ideal pero que no se puede realizar.

COBOL: (COmmon Business -Oriented Language - Lenguaje Común Orientado a Negocios). COBOL es un lenguaje de programación creado en 1960 con el objetivo de crear un lenguaje universal para cualquier tipo de computadora, orientado a la informática de gestión.

ODBC: En informática, el ODBC (Open Database Connectivity) es un estándar de acceso a bases de datos, que permite mantener independencia entre los lenguajes de programación, los sistemas de bases de datos (las bases de datos y su software gestor), y los sistemas operativos.

JDBC: *Java Database Connectivity,* más conocida por sus siglas, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

DBA: Administrador de Bases de Datos

TCP/IP: (Transfer Control Protocol / Internet Protocol). Es el protocolo que utiliza internet para la comunicarse.

LAN: (Local Area Network - Red de Área Local). Interconexión de computadoras y periféricos para formar una red dentro de una empresa u hogar, limitada generalmente a un edificio.

MAN: (Metropolitan Area Network - Red de Área Metropolitana). Red de alta velocidad que cubre un área geográfica extensa. Es una evolución del concepto de LAN (red de área local), pues involucra un área mucho más grande como puede ser una área metropolitana.

WAN: (Wide Area Network - Red de Área Extensa). WAN es una red de computadoras de gran tamaño, generalmente dispersa en un área metropolitana, a lo largo de un país o incluso a nivel planetario.

Login: es el momento de autentificación al ingresar a un servicio o sistema.

Crackers: individuos con altos conocimientos en informática

Hacking: Técnicas y procedimientos utilizados por un hacker para cumplir un determinado objetivo. Suele asociarse esta palabra a procedimientos ilegales o malignos.

Proxies: Un proxy web es utilizado para interceptar la navegación de páginas web por motivos de seguridad, anonimato, rendimiento, etc.

Firewall: (Muro de Fuego - Cortafuego). Herramienta de seguridad que controla el tráfico de entrada/salida de una red.

Caballo: Programa tipo virus que queda activo en el sistema y abre un puerto de entrada a esa computadora. De esta manera la PC queda expuesta y puede ser accedida remotamente.

Holística: La holística alude a la tendencia que permite entender los eventos desde el punto de vista de las múltiples interacciones que los caracterizan; corresponde a una actitud integradora como también a una teoría explicativa que orienta hacia una comprensión contextual de los procesos, de los protagonistas y de sus contextos. La holística se refiere a la manera de ver las cosas enteras, en su totalidad, en su conjunto, en su complejidad, pues de esta forma se pueden apreciar interacciones, particularidades y procesos que por lo regular no se perciben si se estudian los aspectos que conforman el todo, por separado.

ANSI: (American National Standards Institute - Instituto Nacional Americano de Estándares). Organización encargada de estandarizar ciertas tecnologías en EEUU. Es miembro de la ISO, que es la organización internacional para la estandarización

SPARC: (Scalable Processor Architecture). Es una arquitectura RISC originalmente diseñada por Sun Microsystems en 1985. SPARC es una marca registrada de SPARC International, Inc., organización establecida en 1989 para promover la arquitectura SPARC.

Cache: Un caché es un sistema especial de almacenamiento de alta velocidad. Puede ser tanto un área reservada de la memoria principal como un dispositivo de almacenamiento de alta velocidad independiente.

Buffer: (memoria intermedia, intermemoria) Memoria de almacenamiento temporal de información. Suele tratarse de una memoria intermedia entre un dispositivo y otro, por ejemplo, la computadora y la impresora, o la computadora y el disco rígido, etc.

Backup: (Copia de seguridad) Es la copia total o parcial de información importante del disco duro, CDs, bases de datos u otro medio de almacenamiento.

CAD/CAM,CASE,OIS,GIS: (Computer Aided Design - Diseño Asistido por Computadora). Cualquier software que permite hacer dibujos bidimensionales, tridimensionales, y/o técnicos.

XOR: En programación, una condición simple se refiere a aquella que no debe combinarse con otras condiciones para determinarse el resultado de verdad. En tanto, una condición compuesta está formada por dos o más condiciones simples, separadas por los operadores lógicos AND (Y), OR (O), XOR (O excluyente).

Permutación: En matemáticas, dado un conjunto finito con todos sus elementos diferentes, llamamos **permutación** a cada una de las posibles ordenaciones de los elementos de dicho conjunto.

Bits: es el acrónimo de ***B****inary digi****t***. (dígito binario). Un bit es un dígito del sistema de numeración binario.

Mientras que en el sistema de numeración decimal se usan diez dígitos, en el binario se usan sólo dos dígitos, el 0 y el 1. Un bit o dígito binario puede representar uno de esos dos valores, **0** ó **1**.

Desencriptación: Recuperación del contenido original de una información cifrada con anterioridad.

BIBLIOGRAFIA

North, Henry
Fundamentos de Bases de Datos.
México, Ed. Mc Graw Hill, 1998

Date, C.J
Introducción a los Sistemas de Bases de Datos
México, Ed. Addison-Wesly Iberoamericana

Gio, Wiederhold
Diseño de Bases de Datos
México, Ed. Mac Graw Hill, 1985

Martin. James
Organización de las Bases de Datos
México, Ed. Prentice Hall, 1977

James A. Senn
Sistemas de Información
México, Ed.Mc Grawn Hill, 1986

Aceituno Canal, Vicente
Seguridad de la Información
México D.F Limusa c2006

Pino Caballero
Seguridad Informática: técnicas criptográficas
México, D.F Alfaomega c1997

Piattini Velthuis
Tecnología y Diseño de Bases de Datos
México, D.F Alfaomega 2007

Enrique Daltabuit Godas
Seguridad de la Información
México, Limusa, Noriega 2007

Amparo Fuster Sabater
Técnicas Criptográficas de Protección de Datos
México, Alfaomega c1998

García Padilla Carolina
Control de Acceso y Seguridad de la Información Computacional
Tesis de Licenciatura Ingeniera en Computación 2001
FES – ARAGON

Rivera Avalos Pedro
Bases de Datos Orientadas a Objetos
Tesis de Licenciatura Ingeniería en Computación 1996
FES –ARAGON

Paginas de Internet visitadas:

http://www.tierradelazaro.com/public/libros/des.pdf
http://es.Kioskea.net/contents/crypto/des.php3
http://www.cryptotools.com/
http://www.elguille.info/colabora/puntoNET/criptografia/Crypto.htm
http://ccia.ei.uvigo.es/docencia/SSI/Tema3.p2.pdf
http://www.arielorellana.net/des.htm