42
2ej

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

" DISEÑO Y CONSTRUCCION DE UN PROTECTOR DE PROGRAMAS DE APLICACION USANDO TECNICAS DE HARDWARE "

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A:

CESAR HUERTA OLIVARES

DIRECTOR: ING. JESUS RAMIREZ

MEXICO, D. F. 1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

2 CRIPTOGRAFIA

3 DISEÑO DEL PROTECTOR

DIAGRAMA DE BLOQUES

OBJETIVO

Diseñar y construir un protector de programas de aplicación usando técnicas de Hardware. Además de establecer los mejores procedimientos para proteger la información en sistemas de computadoras personales.

INTRODUCCION

Mucho se habla sobre la protección de software, pero a muy pocos les gusta hacer uso de ella. Y quizá la causa de esto sea la falta de conciencia de los usuarios o el desconocimiento de procedimientos adecuados para respaldar su información.

Se piensa que la seguridad de la información es algo difícil y costoso, pero si se compara el costo de la seguridad con los beneficios de la misma se puede observar que este es mínimo.

El tener un exceso de protección puede ser contraproducente y el no hacer nada puede ser peligroso. Entonces, ¿ Cómo se puede saber cuanta protección es suficiente ?

A nadie le agrada el control de accesos mediante códigos secretos, las autorizaciones y los procedimientos necesarios que se tienen al trabajar con los "mainframes" o cualquier sistema multiusuario.

A pesar de que no puede existir una seguridad perfecta, se pretende impedir el acceso sin autorización a los recursos de la computadora, los errores de operación y la pérdida de información causadas por fallas del equipo y daños debidos a incendios y agua. El objetivo de todo esto, no es limitar todo aquello que no podemos impedir. Lo principal, es que cuando se detecte un problema, se recupere la información rápidamente.

Las microcomputadoras a diferencia de los "mainframes" tienen un significado de libertad y simplicidad, no de burocracia en donde existen procedimientos tediosos y complicados.

Desafortunadamente este concepto de libertad no puede ser aplicado en la realidad debido al gran crecimiento del uso de las computadoras personales y de la sensibilidad cada vez mayor de la información. Cuando se refiere a sensibilidad de la información es importante mencionar que cada empresa utiliza la información en forma distinta y por tanto la pérdida de la misma significa diferentes situaciones de conflicto.

Durante el desarrollo de éste trabajo se podrá entender como dependiendo del rubro de cada empresa la información enfrenta diferentes situaciones; por ejemplo, en algunos casos la pérdida de información es vital para la empresa sin que esta sea necesariamente un buen blanco para el plagio, en otros casos, sucede lo contrario y la información que puede ser un buen blanco para el plagio no necesariamente es vital para la compañía.

En una microcomputadora, se debe esperar tener los aspectos básicos de protección en el sistema operativo. Si se analiza el MSDOS se puede observar que este no cuenta con las alternativas de seguridad necesarias. Si bien se puede encontrar en el MSDOS algo de seguridad en forma de códigos de acceso, también se puede encontrar la forma de removerlos con un disco de utilerías fácil de adquirir. Aún el sistema operativo OS/2, el cual ha sido llamado el sistema operativo del futuro, ignora la cuestion de seguridad.

Realmente el diseñar un sistema de protección va más alla del sistema operativo y del manejo normal de operación que realiza un usuario, la seguridad debe involucrar más aplicaciones especializadas tanto de Hardware como de Software, además de una serie de procedimientos que definitivamente estan fuera del alcance de cualquier sistema operativo por mas sofísticado que éste sea.

El diseñar un sistema de protección que sea inviolable es muy difícil, pero aún es más difícil hacer un sistema de seguridad fácil de usar que la gente pueda manejarlo sin tener que recordar un procedimiento tedioso. Además, el hecho de usar el sistema de encripción de archivos más eficiente que exista, llevaría cinco minutos y cincuenta tecleadas del procedimiento para proteger un archivo común; obviamente no faltará la persona que intente violar dicho procedimiento y evitar la "pérdida de tiempo" haciendo respaldos de información.

Actualmente existen en el mercado muchos sistemas de protección que atacan diferentes problemas que atentan contra la pérdida y plagio de información que pueden ser usados para mantener la información "sensible" en forma segura.

En muchos casos se relaciona la protección de software con plagio ó robo de datos y/o programas. Pero no necesariamente es así, y de hecho en la mayoria de los casos el usuario es el responsable.

Dentro del capítulo 1 se explican los problemas a los que se enfrentan ante diferentes conflictos relacionados con la seguridad de la información. Se analizan los problemas que se pueden tener al manipular cualquier tipo de información y de la posible pérdida o destrucción de la misma, sin que necesariamente sea por causa de plagio o de robo.

En éste capítulo se muestran también estadísticas que muestran las formas posibles de destrucción de la información, sin que necesariamente la causa sea plagio o robo. Además, se muestran también estadísticas que pueden ayudar a la selección de procedimientos adecuados para proteger la información sin generar gastos innecesarios que posteriormente pueden ser contraproducentes.

En el capítulo 1 se explica la anatomía de los virus informáticos y de las formas que se tienen para combatirlos. Se mencionan las diferentes vacunas que permiten eliminar la mayoría de los virus conocidos, pero es importante contar con procedimientos y recomendaciones adecuadas para que los usuarios se protejan contra los virus informáticos.

En el capítulo 2 se explican los conceptos básicos relacionados con la encripción de datos mostrando la importancia que esta tiene en el manejo de los sistemas de seguridad.

La criptografía se conoce como el "arte" de hacer lo comprensible incomprensible y viceversa. Esta forma de protección de código es actualmente la más usada en cualquier tipo de transferencia de información. Se explican las ventajas de la encripción para proteger a tráves de códigos secretos.

Es importante mencionar que la encripción es parte esencial del diseño del protector que se presenta en el capítulo 3 y de ahí la importancia que se le da en todo el capítulo 2.

El diseño del protector esta compuesto en gran parte de hardware. La parte de software, sin embargo, juega un papel importante en el diseño del protector.

Si bien se menciona en éste trabajo que un protector creado con software tiene demasiadas carencias, es importante mencionar que complementando las ventajas que proporciona el software con las que proporciona el hardware se puede generar un protector altamente confiable.

Durante el capítulo 3 se explica el funcionamiento de las interrupciones en las computadoras personales y el manejo de programas residentes en memoria (TSR). Estos conceptos forman la parte especializada que conforman la parte del protector realizada en software. En éste capítulo, se explica también el manejo de discos flexibles a tráves del ROM-BIOS y de las formas de acceso por medio del manejador de discos flexibles.

La parte correspondiente al hardware del protector lo forma el secuenciador microprogramado de AMD (Advanced Micro-Devices) y el diseño de la arquitectura de este procesador Bit-slice.

Al final de éste capítulo 3 se explicará el funcionamiento del diseño del protector por medio de los conceptos que previamente se formularon.

Se darán conclusiones y posibles aplicaciones adicionales utilizando la arquitectura presentada en este trabajo explicando las ventajas y desventajas de ésta configuración. En la parte final se incluyen apendices generales, del circuito protector y de los programas utilizados.

Por último quiero agradecer a todos aquellos que de alguna forma ayudaron a enriquecer el presente material y en especial al Ing. Jesús Ramírez por su apoyo en la elaboración de éste trabajo.

César Huerta Olivares

1 SEGURIDAD DE LA INFORMACION

La información es el activo más valioso en cualquier tipo de organización, y por tanto, se deben de tomar las precauciones adecuadas para evitar su destrucción o el uso inadecuado que se le de a la misma. Existen varias formas recomendables para protegerla, y es decisión de cada organización, establecer las normas básicas de seguridad que le permitan una adecuada fluidez en el uso de la información y que al mismo tiempo se verifique su integridad.

La seguridad absoluta es inalcanzable. Existen muchas alternativas comerciales con altos grados de seguridad, pero estos pueden no ser los apropiados. Cuando se evaluen varias opciones, se debe considerar cuanta seguridad se necesita realmente en base a nuestras necesidades.

En muchos casos, un sistema de código de acceso de entrada puede ser suficiente tanto en "hardware" como en "software". A pesar de que un sistema de este tipo puede ser sencillo de violar por cualquiera que cuente con el conocimiento necesario para hacerlo, puede existir la ventaja de que no hay que preocuparse si se olvida el código de acceso. Aquí se puede considerar que el tipo de ladrones contra los que se protege la información no tiene los conocimientos técnicos suficientes o la paciencia necesaria para violar un sistema de seguridad de este tipo.

Cualquiera que sea la situación, es importante evaluar el riesgo. Si el riesgo que se tiene proviene del exterior, la encripción y el control de accesos físico puede ser más útil; si el problema por el contrario es interno, otros métodos menos sofisticados pueden dar resultados excelentes. No hacer nada puede ser peligroso, pero, un esquema de protección demasiado elaborado es igualmente irresponsable, debido a que nadie lo usará ni podrá adaptarse a éste. La complacencia y el pánico son igualmente dañinos, por lo cual, una apreciación razonada es esencial.

Actualmente los llamados virus informáticos han provocado que la seguridad de la información sea de la atención pública. Sin embargo, este tipo de atención esta dirigida hacia los ataques premeditados contra la información,y se esta perdiendo la atención a los errores provocados por usuarios autorizados durante su trabajo diario.

En este capítulo se mencionarán algunas de las muchas opciones que se tienen para proteger la información de acuerdo a las diferentes amenazas en contra de las misma.

SEGURIDAD DE DATOS

Definición .-

Es importante poder definir el concepto de seguridad de la información, para evitar caer en posibles confusiones, de acuerdo a su estudio, aplicaciones y alcances. La definición más completa que se puede dar de seguridad es la siguiente :

" *La seguridad de la información se refiere a la protección de los activos de información contra revelación, modificación, o destrucción accidental o intencional no autorizada.* "

SEGURIDAD BASICA

Actualmente la información juega un papel primordial en el desarrollo industrial y comercial, y por tanto, en el sector económico. Los sistemas de computo se hacen cada vez más accesibles a más usuarios. Este crecimiento provoca que se tengan inadecuados sistemas de seguridad que permitan proteger la información en los sistemas de cómputo, y al mismo tiempo proporcionen libertad en el manejo de la información de los sistemas automatizados.

Los diferentes sistemas de protección que existen actualmente en el mercado pueden ayudar a mantener segura la información, pero al mismo tiempo pueden tener puntos debiles en donde nuestra información es vulnerable.

Actualmente existen muchos protectores que pueden aplicarse a la protección de información, pero antes de la elección de uno se tiene que realizar una evaluación adecuada para que los procedimientos normales de trabajo se alteren lo menos posible. En pocas palabras, se debe evitar la burocracia al manejar la información para que ésta fluya adecuadamente.

También se debe tener cuidado de no instrumentar la seguridad en exceso, ya que esto puede provocar que se descuiden otras áreas que pueden ser igualmente sensibles.

Cuando se realiza la evaluación, se debe considerar en primer término contra quien se protege la información, es decir, cuales pueden ser las posibles fuentes que afecten la seguridad en los sistemas.

Posiblemente al realizar dicha evaluación nos encontremos que no es necesario tener el protector de información más sofisticado y que es suficiente seguir algunas reglas de seguridad de sentido común.

Algunas de estas reglas de seguridad son :

- Regla cardinal :

  Si la información que se maneja es sensible y se usa un disco duro, se debe evitar dejar almacenada ésta en la máquina. Se debe copiar la información a un disco flexible y mantenerla bajo llave.

  Se recomienda no usar las utilerías del sistema operativo para borrar los archivos del disco duro ya que el ERASE/DEL del MS-DOS no elimina los archivos del disco, y estos pueden ser fácilmente recuperables y accesados.

  Se deben nombrar los archivos que contienen información importante con nombres que no indiquen su posible aplicación o contenido, es decir, si se tiene un archivo con información del presupuesto para un determinado año, se debe evitar nombrarlo PRESUP.ANL. En estos casos se recomienda darle un nombre ilógico y tener un directorio aparte que se encuentre inaccesible a cualquier usuario. Posiblemente ningún ladrón de información se detendrá ante éste obstaculo, pero evitamos tentaciones.

En pocas palabras, se debe evitar lo obvio para evitar dar facilidades que provoquen que la información sea corrompida.

- Usar todos los dispositivos de seguridad existentes :

  Si se tiene una cerradura en la oficina, lo más conveniente es usarla y no dejar la información accesible.

- Checar los desechos de impresiones :

Generalmente las impresiones pueden proporcionar el mejor medio de acceso a información confidencial, principalmente en las oficinas en donde las impresiones se utilizan como forma primaria de respaldos. Si la impresora que se utiliza usa cinta de carbón, se debe tener cuidado, ya que ésta puede ser releída e interpretada.

Se debe desechar la información confidencial y asegurarse de que el área de impresion quede limpia. Si esta no se limpia, alguien puede imprimir lo que se encuentre en el área de impresión y accesar datos confidenciales. Es necesario mencionar que esto ocurre generalmente en todos lados y que las acciones que se deben tomar son de tipo preventivo.

- No usar claves de acceso obvias :

Las personas que se dedican a violar los sistemas de seguridad saben que los usuarios de computadoras utilizan claves de accesos relacionados con la persona. Estas claves de acceso pueden ser las iniciales de la persona, el nombre de algún hijo, la fecha de nacimiento, etc.

- No grabar la claves de acceso bajo el teclado o cerca del área de trabajo :

Es importante el evitar escribir la clave de acceso en lugares en los que se pueda recordarla fácilmente. El usuario debe buscar otras forma de obtener su clave de acceso en caso de no recordarla. Si en el peor de los casos se tiene que grabar en algún lugar cercano del área de trabajo, se recomienda que no se grabe la clave en forma exacta.

- Use el sentido común :

Posiblemente la información con la que se trabaje no sea tan "importante" para algunas aplicaciones. A menos que ésta información sea altamente

sensible, se recomienda usar los procedimientos de sentido común que pueden ser adecuados y suficientes.

Riesgos de seguridad .-

Aún cuando se tenga el conocimiento real de las posibles amenazas, todavía se necesita saber que acciones tomar y cuales están justificadas. La protección contra cualquier tipo de amenaza tiene sus riesgos, y la determinación de ese riesgo es el punto clave para saber que acción tomar.

Conforme se instrumentan los controles de seguridad, el costo que se agrega a dichos controles aumenta al mismo tiempo que el costo de pérdidas esperadas decrece. El objetivo en estos casos es no gastar más dinero en un determinado período que el esperado en pérdidas si no se toma niguna acción.

Existen tres opciones disponibles para manejar el riesgo :

- **Evitarlo.-** Esto es, mediante la instrumentación de controles de seguridad.

- **Asignarlo a otros.-** Esto es, generalmente un seguro contra interrupción del negocio.

- **Asumirlo.-** Si el costo esperado es menor que el costo de la acción a tomar, entonces, el no tomar una acción sería lo adecuado.

Algunas de las amenazas a las que nos enfrentamos se encuentran acontinuación. Esta lista se basa en un estudio original realizado por el gobierno de Estados Unidos desarrollado hace diez años.

1. Errores y omisiones. Se estima en un porcentaje de pérdida financiera de 50-80 %.

Causas : estress, falta de entrenamiento, mala supervisión, aunque los empleados honestos son el principal objetivo de estudio.

2. Empleados deshonestos (15-30%). Acciones accidentales que benefician a un empleado, generalmente detectado por un error y no detectado administrativamente.

3. Fuego y desastres naturales (10-15%).

4. Empleados disgustados (5-15%)

5. Agua, no necesariamente proveniente de desastres naturales (por abajo de 10%).

6. Otros, (no más de 5%). Esto incluye plagio externo.

Políticas y organización .-

El primer paso, y la acción simple más importante que puede ser tomado para comenzar un programa de seguridad de la información, es la formulación de una política de protección de información , la cual se firma por el director general y que se debe aplicar a todos los empleados. Esta política debe ser corta, de una a cuatro páginas, y ser difundida hacia todas las áreas operativas. Además de la política, se deberán generar documentación con estandares, guías, y procedimientos.

La política debe contener la definición que la organización de a la seguridad de información. Esta debe especificar los derechos de propiedad, clasificación de la información, las responsabilidades individuales de administración, dueños de la información, usuarios e información del grupo de seguridad. Esta política no debe ser técnica, pero debe contener detalles sobre identificación personal, y control de claves.

Educación y Programas de seguridad .-

Sin una adecuada educación en política de seguridad de información para los empleados, el programa de seguridad no podrá funcionar eficazmente. Es importante que todos los individuos entiendan en forma clara que activos están bajo su responsabilidad y que tipo de protección deben recibir esos activos.

Se recomiendan seminarios periódicos que expliquen la importancia de la información. Es importante la orientación a los nuevos empleados ya que estos pueden ser inducidos más fácilmente.

Controles de acceso .-

Lo más importante en cualquier sistema de seguridad son los controles físicos. Si los controles físicos no son adecuados, se debe entonces pensar en gastar en otros sistemas sofisticados de seguridad.

Cuando se habla de controles de acceso se debe de meditar profundamente sobre la eficiencia de los mismos. Muchas veces se puede encontrar que los controles de acceso impiden el manejo libre de una computadora personal y de sus recursos. Además, estos controles van en contra de la filosofía con la que las computadoras personales fueron originalmente diseñadas.

Es muy importante que cuando se quiera usar este método de protección se tomen en cuenta las siguientes cuestiones :

Qué tan efectivos son los controles de acceso dependiendo del tipo de información que se maneja y la trascendencia de la misma ?.

En donde se pueden aplicar ?.

Se tienen otros métodos de solución alternativos ?.

Al hablar acerca del tipo de información y de la trascendencia de la misma, se hace referencia a que la información se esta convirtiendo en uno de los activos más importantes de muchas empresas. Quizás la información que se genere a nivel académico en las universidades no sea tan sensible y se puedan adoptar medidas de seguridad que no interrumpan el máximo aprovechamiento de los recursos de computo.

Los controles de acceso se usan actualmente en equipos grandes y estos han dado un gran resultado y obviamente las ventajas son superiores a las desventajas.

El sistema más simple de código de acceso que manejan las microcomputadoras es a nivel del arranque de la máquina. Al iniciar el proceso de arranque la máquina interrumpe su proceso para solicitar el código de acceso y validarlo. Cuando el código de acceso es correcto, el proceso de arranque continua y el acceso es permitido. Un método común es introducir una linea como DEVICE = PASSWORD.SYS en el archivo CONFIG.SYS para activarlo. Un sistema tan simple como este tiene muchos defectos. Quizás el problema más significativo sea que la computadora quede vulnerable una vez que el proceso de arranque se ejecuto satisfactoriamente. Sin embargo, el elaborar un sistema de código de acceso más sofisticado en la microcomputadora tendría los mismos problemas, debido a que sería atacado de la misma forma.

Tratar de adivinar un código de acceso es imposible. Aunque se puede apagar la computadora cuantas veces sea y repetir el código de acceso de entrada, las posibilidades de éxito son escasas, a menos que el código de acceso sea muy obvio.

Sin embargo, I.B.M. y Microsoft venden actualmente un disco eficiente para romper sistemas de código de acceso que no utilizan encripción como los mencionados antes : I.B.M. lo llama PC-DOS y Microsoft MS-DOS. Lo que se tiene que hacer es comprar el disco, insertarlo en el drive A, encender la máquina y el acceso a toda la información de la computadora estará listo.

Realizando el proceso de arranque desde un disco flexible que no tenga los archivos CONFIG.SYS y AUTOEXEC.BAT, se puede violar cualquier protección de código de acceso y accesar y alterar información del disco duro. Si la computadora utiliza un "hardware" no estándar y manejadores de dispostivos especiales, quizás tome un poco más accesar la información, pero es igual de simple.

El problema con la mayoría de los sistemas de código de acceso es que estos no son del todo seguros. A diferencia de los "mainframes" que son físicamente inaccesibles, las microcomputadoras se encuentran en las oficinas, físicamente accesibles y muchas de ellas son actualmente portables. Se puede decir que los códigos de acceso para microcomputadoras es equivalente a construir puertas fuertes con chapa de seguridad para una caja de cartón.

La restricción de accesos es un prerequisito del control de accesos. Las restricciones pueden tomar diversas formas. Los controles pueden ser físicos, como cuando se tiene una computadora en un cuarto bajo llave en el cual no sería necesario el uso de "hardware" o "software" adicional como medios de protección; otro control sería por software, como son los códigos de acceso y finalmente por hardware, como puede ser una chapa que impida encender la microcomputadora.

Los controles de acceso funcionaron bien en el pasado, cuando existían pocas microcomputadoras y habían pocas personas que conocían su funcionamiento, pero, actualmente ya no es así.

La restricción de acceso puede ser lógica, instrumentada por "hardware" y "software". La encripción ofrece la forma de hacerlo más sencilla en cualquier forma. Obviamente esta es la forma recomendada si se opta por un control de accesos.

Usando datos encriptados con un programa desencriptor puede parecer similar a usar un sistema que maneje códigos de acceso, y de hecho para el usuario no existe diferencia alguna, quien posiblemente no aprecie la diferencia entre un código de acceso y una llave de encripción, pero en realidad no es así. La diferencia básica es fácil de apreciar; si se deshabilita o se viola el sistema con código de entrada , todos los recursos en la computadora se vuelven accesibles, y por otro lado si se logra burlar el programa de encripción, no se podra leer toda la información ni utilizar todos los recursos de la PC.

Con ésto se puede demostrar un punto muy importante y que se centra más en lo que es la protección de software. Lo que realmente se esta protegiendo no es la computadora, sino la información que esta es capas de almacenar o manejar; y esto es algo que se debe de considerar en primer lugar cuando se escoja un sistema de protección que utilice la encripción como medio principal.

Para asegurar la protección de la información se debe considerar no solo la prevención de accesos no autorizados, sino la posibilidad de pérdida de información. La prevención no autorizada de accesos se lleva mejor acabo no teniendo respaldos, por otro lado, para evitar la pérdida de información se requieren tantos respaldos como sean posibles. La encripción puede resolver esta contradicción de una forma muy simple y efectiva. Si los respaldos en disco son encriptados en forma segura, se puede hacer un número ilimitado de respaldos sin comprometer información confidencial.

Con la gran difusión de las computadoras personales, no solo se debe poner atención en la protección usando medios físicos, sino que además se debe poner especial atención en la protección en contra del plagio y del ataque intensionado contra la información mediante el uso de "software" o "hardware".

Clasificación de la información .-

Aunque se necesita clasificar la información en cuanto a revelación, modificación y destrucción, es en este último en donde debemos poner la mayor atención. La clasificación de la información es simplemente una etiqueta que le indica a la gente en la organización que tipo de controles se requieren basados en la política de su organización y en una serie de reglas específicas, de aqui la importancia de la educación. Las alternativas para una política de clasificación de la información son, ó proteger todo, ó no proteger nada, pero ambas son igual de costosas, y es aqui donde la evaluación juega un papel primordial.

Algunos programas en particular son muy sensibles a modificaciones. Otra información es sensible a la destrucción; en éste caso la destrucción evitaría que la compañía siguiera con su misión. Es importante mencionar que estas dos clasificaciones junto con la basada en sensibilidad a la revelación, son generalmente mutuamente excluyentes; el hecho de que un registro sea sensible no necesariamente implica que la información tambien sea confidencial, por ejemplo, las cuentas de gastos de una organización.

Recursos de cómputo .-

Se consideran recursos de una computadora a la información que se almacena o procesa, el hardware y software.

El hardware se refiere a todo el equipo físico que forma parte de la computadora y a sus dispositivos periféricos, ya sean estos de entrada o de salida. El software, se refiere, a todos los programas de sistema o de una aplicación en particular. Sin embargo, el recurso que se puede considerar más valioso, son las personas que usan y conocen el sistema de información.

Para tener un adecuado sistema de seguridad es necesario referirse a los recursos de la compuradora como parte del activo de la organización. de tal forma que se evalue el alcance de los procedimientos a instrumentar.

Es conveniente , además, que se identifique claramente el rubio de la compañía. Esto con el fin de identificar el activo más importante entre la información, el hardware o el software; por ejemplo, en una compañia dedicada a fabricar computadoras, el recurso más importante sería el hardware, pero, en una compañía en donde la integridad de la información es lo más importante, el hardware pasa a ocupar un segundo término.

Ahora bien, como se menciono antes, la gente usuaria de la información y de los sistemas de información deben considerarse como el recurso más importante en una empresa. Esto se debe principalmente a que ellos son los responsables de darle forma a los datos que se almacenan y manejan a tráves de un sistema de cómputo.

AUDITACION Y CONTROLES

Administración de controles .-

El uso de una adecuado manejo de controles es la forma más simple de mantener un buen programa de seguridad de información.

A continuación se dan recomendaciones relacionadas con el manejo de controles.

a. Separación de responsabilidades

b. Los cambios a un sistema se deben aprobar y realizar en forma independiente.

c. Los controles deben existir para asegurar que todo el trabajo sea procesado de acuerdo con los procedimientos.

d. Los controles deben existir para una periódica detección de errores.

e. Los procedimientos personales, tales como la rotación de empleos, prácticas de recesión y de contratación, deberán ser revisadas para evaluar su impacto en relación a la seguridad de la información.

Auditorias de revisión.-

Además de las auditorias internas y externas, se debe contemplar un programa de auto-revisión o "auditoría amigable" con el proposito de reparar los problemas y no de ser reportados simplemente a la dirección.

Auditorias externas e internas.-

Las auditorias formales deben incluir seguridad de la información tanto de medios de accesos físicos como lógicos. Se deben aplicar pruebas a los controles en forma aleatoria sin previo aviso. Tanto las auditorias externas como internas deben ser medios de la dirección para asegurarse que las políticas se esten aplicando correctamente.

PLANES DE CONTINGENCIA

Documentación del plan de contingencia.-

Se debe tener un plan en el cual se proporcionen las intrucciones para continuar con las aplicaciones críticas en el caso de un evento catástrofico. Un desastre o catástrofe puede afectar muchas partes de la organización, de ahí la importancia de la participación de toda la gente potencialmente afectada.

El objetivo del plan debe estar dirigido a funciones críticas , aquellas necesarias para mantener funcionando la organización, aún cuando se requiera moverse a otras localidades.

El plan debe ser distribuido a los ejecutivos y a los administradores responsables junto con su copia que se deberá guardar en un lugar externo a las instalaciones normales.

Las responsabilidades individuales deberán ser descritas en detalle. Deberá incluirse los detalles de equipo y materiales para que se pueda continuar con las labores en extra-muros.

Respaldos de información .-

Es necesario establecer un plan de respaldos periódicos con el fin de proporcionar seguridad en la información vital para la empresa. Se debe contar para esto con un lugar de almacenamiento que cuente con las condiciones ambientales y de construcción que garanticen la seguridad de la información. También se debe de instrumentar un plan para la auditación de la información que se guarde en forma externa para hacerlo en forma periódica.

Responsabilidades del usuario.-

En el plan del usuario es necesario incluir los procedimientos que este necesite para evitar la interrupción de sus labores. Además como parte importante, se debe incluir los,procedimientos administrativo, no automatizados, que este debe seguir.

El usuario debe encargarse de organizar los recursos que debe tener consigo para poder actualizar su información en extra-muros si esto fuera necesario.

Pruebas e instrumentación .-

Para que sea efectivo, todas las partes del plan deben ser verificadas regularmente. Los planes de emergencia deben ser probados. Se debe probar la información almacenada en forma externa, así como las formas de transporte que se deberán usar para la información en caso de una contingencia. Los resultados de las pruebas deberán ser documentados y con recomendaciones para efectuar posibles cambios en los procedimientos cuando sea necesario.

Es importante evaluar las instalaciones extra-muros y verificar que cuenten con los recursos necesarios para instalar equipos electrónicos y mobiliario. Para esto, es necesario realizar una prueba practica que garantice una operación "normal" para el usuario.

CODIGOS DE PROTECCION

Cualquier buen sistema de seguridad debe tomar en cuenta la encripción, considerando que los estandares DES y RSA proporcionan una buena base para un sistema de seguridad de computadoras, ya que como se explicará más adelante el mantener oculto el algoritmo de encripción no implica mayor seguridad y conducen a un sentido falso de lo que se debe entender por seguridad.

Pero la encripción por sí sola no es suficiente. El seleccionar la llave adecuada, el manejo de la llave, la seguridad física, la seguridad de la gente y los procedimientos para asegurar que el "plaintext" ó código fuente no se "escape" del sistema por algún pretexto es esencial para un sistema de seguridad de información en computadoras personales.

ENCRIPCION DE DATOS

Se afirma, y con razón, que cualquier sistema de seguridad adecuado, debe depender en un mayor porcentaje de la encripción. Se habla de un mayor porcentaje para la encripción porque ésta por si sola no es suficiente, pero es necesaria en cualquier buen sistema de protección.

La criptografía es el antiguo arte de hacer lo comprensible, incomprensible. El uso de los códigos secretos se remonta a la época de Julio César, cuando este protegía sus mensajes secretos remplazando cada letra en el texto original, llamado "planitext" con una letra tres posiciones después en el alfabeto.

El texto destino se llama "ciphertext", en el cual A esta representada por D, B por E, y así sucesivamente.

La guerra entre los expertos en criptografía y los decifradores de código, se ha incrementado desde que se inventó la computadora. Por un lado, las computadoras ayudan a descifrar sistemas de criptografía complicados en unos cuantos segundos; y por otro lado, han hecho posible el uso de algoritmos de encripción extremadamente complejos que eran anteriormente impracticos. Más aún, los actuales sistemas distribuidos de computación, la gran disponibilidad de las microcomputadoras, los avances en almacenamiento masivo, y el diverso uso de las comunicaciones por computadora han contribuido todos ellos a desplazar la criptografía de la milicia, los aspectos diplomáticos y gubernamentales a otros campos de importancia general.

Actualmente se usan los sistemas de encripción convencionales y los públicos. Estos sistemas estan representados por el DES (Data Encryption Standard), en el caso de los convencionales, y el RSA (las siglas correponden a las iniciales de sus inventores), para el caso de los públicos.

VIRUS INFORMATICOS

Antecedentes .-

Ciertos programas que son capaces de destruir información y en algunos casos más complejos hasta causar daños físicos y que además son capaces de reproducirse se les llama virus informáticos.

El antecedente de los virus se remonta en los años 50's cuando el norteamericano John Von Neuman los llamo programas que se autoreproducen.

Al principio se consideró a los virus como simple diversión, hasta que en el 2 de noviembre de 1988, dos redes norteamericanas muy importantes fueron infectadas. Dicho virus, causo daños en más de 6000 computadoras de centros privados, instituciones académicas, instituciones militares y gubernamentales. Este incidente a despertado gran alarma entre los usuarios de sistemas personales pues se ha mostrado la vulnerabilidad de los sistemas de información.

El principal medio de contagio que los virus informáticos toman para atacar las computadoras personales, es a través del uso de un disco flexible infectado.

Actualmente estos virus se estan volviendo más benignos y sofisticados; algunos se pueden ocultar perfectamente por si mismos que ni aún las mejores utilerias pueden detectarlos. Pero, además de ésto, existen virus que pueden ocultarse en memoria no volátil, provocando que el contagio se produzca y el problema se vuelva cada vez mayor.

Las formas de infección más comunes es a través de las llamadas o accesos de entrada/salida. Son en los accesos de éste tipo en donde la mayoría de los virus conocidos realizan su contagio en forma transparente al usuario.

Aunque exista una gran controversia con los virus informáticos, existe cierta información básica que se debe tener para poder conocerlos mejor y poder enfrentarlos de forma adecuada.

Los virus informáticos están compuestos de un código que infecta otros programas, se reproducen por si mismo y requieren de un segmento residente en disco para almacenarse. Una vez de que el virus se encuentra físicamente en el sistema, este puede dañar o destruir la información, el medio ambiente, el mismo sistema, y a los propios periféricos conectados.

Dependiendo de la función que realizan, la forma en como se presentan y el daño que pueden causar, es como se les da el nombre a algunos de los virus.

Los virus informáticos presentan relativamente un nuevo problema en cuanto a seguridad de la información se refiere. Es por eso que estos no deben mantenerse ignorados cuando se hable de un plan de seguridad.

Los virus informáticos en las organizaciones.-

Los eventos que se relatan a continuación explican la forma en que un virus se introduce en una organización y luego se expande dentro de esta. Supóngase que la organización contrata una nueva persona para que esta desarrolle cierto trabajo. Parte de este trabajo involucra el manejo de una computadora personal. La persona trae por su parte su procesador de texto favorito de tal forma que le ayude en su nueva asignación.

Seguramente la persona no sabía que su procesador de texto estaba infectado con un virus. El uso de este procesador de texto en una de las máquinas de la compañía va a provocar que el virus se propague en forma masiva, posiblemente a través de una hoja de cálculo o de otro procesador de texto. De esta forma el nuevo virus se encuentra en la organización y su etapa destructora no tardará en entrar en acción.

El virus que se introdujo a la compañía por el nuevo empleado permanecerá dentro de ésta inclusive una vez que el empleado deje la compañía. Un virus puede propagarse por diversas formas a tráves de programas de aplicación o archivo de datos, y puede ser transportado hacia otras localidades por medio de un disco flexible o de una red de datos.En pocas palabras, una copia de cada virus puede hacer múltiples reproducciones de si mismo, y puede infectar cualquier programa o datos que accese en un tiempo muy corto.

Una vez que la organización hace el compromiso para combatir los problemas de los virus en las computadoras, existen diferentes áreas que deben recibir atención. Esto se debe hacer con el fin de evitar que algún virus penetre en la información y la destruya.

Infección de Virus .-

Existen muchas maneras de que un virus infecte un sistema. Cada vez que un programa corre y altera otros programas, existe la posibilidad de infección de un virus. Cada vez que un usuario ejecuta un programa que fue escrito por otra persona, compilado o ligado con alguna librería la información esta en peligro de ser infectada, además, los recursos a los que ese programa tiene acceso quedan en posibilidad de ser infectados y de propagar el virus. Por tanto, el virus cuenta con las "manos" de cada persona que contribuyo a la creación ese programa, ese compilador, o esas librerías.

La introducción adicional de un programa infectado puede ocurrir a través de una gran variedad de canales incluyendo :

- Software introducido por una persona externo que tuvo acceso al sistema.
- Software usado en la casa por un empleado cuyo sistema, sin conocimiento de empleado, esta infectado.
- Software intencionalmente infectado por empleados mal intencionados o disgustados.
- Cualquier tipo de software transferido o compartido con en la organización, o entre la organización y alguna fuente externa.

Estos son los puntos que se recomiendan seguir para la creación de una política que permita combatir y erradicar los virus.

Identificación de virus.-

Existen muchos tipos de amenazas para la seguridad de la información. La amenazas para el rastreo de llamadas internas son altamente reducidas con el sistema de retrollamada. Las amenazas por empleados inconformes puede ser rastreada por una persona responsable. Una característica importante que hace un que virus sea distinto de otras amenazas, es la dificultad de localizar el origen que le dió entrada en la organización.

Los virus, a diferencia de.otras amenazas, atacan en forma inesperada modificando la información .También en otros casos los virus se mantienen estáticos, sin dejar de contagiar, hasta cumplir una cierta condición que los hace comenzar el proceso para el cual fueron diseñados. La variedad de efectos posibles que puede tener un virus al "dispararse" es lo que en algunos casos causa confusión. El término "virus" es algunas veces mal utilizado para referirse a cualquier cosa indeseable que le puede suceder a la computadora. Esto en algunos casos puede hacer que la organización no se encuentre preparada para reaccionar ante un verdadero virus que produzca daños.

Un sistema de cómputo no puede ser infectado si se corre sólo en un disco duro y no se cargan o crean programas o datos en éste. Pero esto no es práctico en muchas circunstancias. Como en otros aspectos de seguridad, se deben medir los beneficios, funcionalidad, y riesgos potenciales, y después luego tomar la acción costo-beneficio para poder ayudar a controlar dichos riesgos. A pesar de las medidas preventivas, es prudente el anticiparse a un ataque por parte de un virus. Debido a esto, la detección de un virus es un componente importante de la seguridad de la información.

Los dos recursos disponibles para la detección de virus son los usuarios y los programas detectadores. Los usuarios deben estar al cuidado de una posible infección, tal como lo hacen para los respaldos de información, y conocer las características de las cuales deben tener cuidado. Adicionalmente, se puede hacer uso de utilerías que están disponibles para ayudar al usuarios o al grupo de soporte a detectar o erradicar posibles infecciones.

Debido a la rapidez con que un virus se puede propagar, es importante que se detecte tan pronto como sea posible. Si un virus se logra propagar en forma incontrolada, es difícil predecir las consecuencias.

En la mayoría de los grupos de usuarios, es posible identificar a los que hacen una gran cantidad de intercambios de información, quienes generalmente corren nuevo software antes que la mayoría lo haga, y quienes además, son los primeros en usar nuevos paquetes. En los sistemas multiusuario, algunas de estas gentes tienen autorizaciones y privilegios especiales que "aprovechan" para probar software de dudosa procedencia. Cuando algún virus se introduce por medio de estos usuarios, éste se puede propagar en forma rápida a otras áreas. Se deben hacer decisiones de beneficio/costo sobre cuales usuarios deben "gastar" el mayor tiempo y esfuerzo en la detección de virus y de esta forma se les entrene en forma especial.

POLITICAS DE SEGURIDAD CONTRA VIRUS

Educación del usuario .-

Las buenas políticas de seguridad dependen del conocimiento y cooperación de los usuarios. Los usuarios deben estar prevenidos de los riesgos que existen y conocer que hacer en caso de que se sospeche de algún problema de seguridad. En forma particular, deben saber a quien llamar si se tienen dudas o sospechan de algún problema, y deben saber que hacer y que no hacer para minimizar posibles riesgos.

Se recomienda una estrategia de prevención y detección de virus informáticos. Una parte importante de esa estrategia es que los usuarios conozcan un procedimiento si es que se detecta la presencia de un virus. En estos casos se recomienda la educación de los usuarios finales, la gente de soporte de primer nivel, y la gerencia involucrada a todos niveles antes de que ocurra algún problema para que se tomen las acciones necesarias de recuperación de información.

Respaldos contra virus .-

Aún sin existir la amenaza de virus, los respaldos adecuados son parte importante del manejo de los sistemas. Cuando un programa o algún archivo de datos se pierde, una buena cantidad de respaldos es esencial. El daño potencial que puede causar un virus sólo aumenta la necesidad de contar con procedimientos adecuados para respaldar la información.

Aunque los respaldos son necesarios para recuperar la información, esto pueden representar un lugar para que se aloje un virus. Se debe tener mucho cuidado para no reintroducir un virus durante el proceso de respaldo de información. Todos los respaldos deberán ser inspeccionados para asegurarse de que no se encuentra presente algún virus en un respaldo. Es necesario ser cuidadoso al restaurar sólo información que no ha sido infectada o cambiada por el virus.

Riesgos de los virus .-

Los virus se pueden propagar a través de un proceso normal de negocios de un usuario a otro en un sistema simple, y a su vez de un sistema a otro. Un virus puede penetrar a la organización siendo creado dentro de ésta o traído de forma externa. Aunque un virus no puede ser creado accidentalmente, un virus puede ser introducido a la organización en forma intencional o no intencional.

Existen dos formas principales para reducir el riesgo de una infección y la propagación de una infección existente :

- La limitación de acceso a los sistemas, y
- La limitación de la disponibilidad de las funciones en el sistema.

Limitación de accesos al sistema.-

En las organizaciones se deben evitar los accesos a los sistemas de las personas externas a ésta. Un ejemplo es la limitación de las personas externas para transmitir archivos ejecutables, o para tener acceso interactivo, total a los sistemas internos.

De forma similar, el movimiento de programas entre computadoras personales a través de discos flexibles puede causar una infección de un sistema a otro. En este caso, es necesario que se apliquen políticas para que los empleado no usen información que sean traídas de fuentes externas, o que por lo menos apliquen herramientas para la detección de virus.

Limitación de las funciones en el sistema.-

Cuando sea posible, se debe limitar a los usuarios en la capacidad para agregar o cambiar programas en los sistemas que estos usan. Idealmente, esta capacidad debe ser restringida para uso de personal autorizado con alto entrenamiento en detección y eliminación de virus. De esta forma, si no se introducen nuevos programas, es imposible que se introduzcan nuevos virus en los sistemas.

Políticas de protección para librerías de software.-

Las bibliotecas de software son lugares en donde residen los programas y que son usados por un gran número de personas. Estos pueden ser discos o un "mainframe", los cuales pueden ser accesados por varios usuarios con cuentas distintas. Estas pueden ser también discos en un archivo servidor de una red de área local, de la cual muchos usuarios obtiene programas comunes.

Estas librerías se encuentran ante un alto riesgo ya que son usadas en forma común y pueden ser fácilmente infectadas. En estos casos, y ante el riesgo de una propagación rápida, las políticas son de gran ayuda para reducir los riesgos.

Por ejemplo, se puede establecer una política que obligue a que cada vez que se agregue un programa a alguna librería esta sea checada por personal calificado para que identifique una posible infección.

Cuando los usuarios tienen contacto frecuente con información sensitiva, sería prudente el establecimiento de políticas más estrictas. Los programas deberán ser probados en un ambiente aislado para poder detectar posibles problemas sin afectar la información productiva.

Existen otras áreas dentro de la organización que pudieran propagar un virus rápidamente, y que por tanto deben ser controladas cuidadosamente. En general, cualquier lugar en el que se distribuya software en forma masiva es potencialmente una fuente de infección.

Políticas de protección para el desarrollo de sistemas.-

El desarrollo de sistemas en considerado como la generación de programas para distribución interna en la compañía, o para venta a clientes.

Debido a que la infección durante el desarrollo de sistemas puede tener serias consecuencias, se debe tener especial cuidado al protegerlos y al definir las políticas de cambios.

Hasta donde sea posible, el desarrollo de sistemas se debe hacer aislado a los sistemas productivos. Cada archivo que se almacene debe ser checado en contra de cualquier infección posible.

Sintomatología de los virus.-

Si los usuarios se fija en los síntomas visibles que presenta un sistema infectado, estos usuarios pueden servir como una importante línea de defensa. Los usuarios deben conocer el comportamiento en una computadora cuando se encuentre presente algún tipo de virus, y además debe saber a quien reportar dichos problemas.

Debido a que los errores de software y problemas de hardware son más comunes que los virus, es importante evitar rumores infundados sobre posibles infecciones, ya que esto puede ser realmente contraproducente. Sin embargo, una infección real requiere una acción rapida. De esta forma se recomienda un análisis profundo para eliminar las "falsa alarmas", que como se menciono antes pueden causar serios problemas.

Algunos de los sintomas que han sido detectados por investigadores de IBM se listan a continuacion:

- Cambios inesperados en las fechas de actualización o en la longitud de los archivos. Particularmente en archivos ejecutables.
- El tiempo de inicio en la ejecución de un programa es mayor y en otros casos el tiempo de corrida es menor.
- Algunos programas intentan escribir sobre medios protegidos contra escritura sin razón aparente.
- Se presenta inexplicable reducción en la capacidad de memoria, o incrementos en el número o tamaño de las areas marcadas como "dañadas" en los medios magnéticos.
- Los archivos marcados como ejecutables se "pierden".
- Se presentan reinicio total al sistema.
- Se presentan cosas inexplicables en la pantalla, incluyendo pelotas rebotantes y mensajes malignos.
- Se presentan cambios en las etiquetas de volumen en los medios de almacenamiento secundario.

- Cargas poco comunes en las redes de área local u otros medios de enlace, especialmente cuando se envían múltiples copias de la misma información al mismo tiempo.

Es importante recordar que futuros virus se comportaran en forma distinta. Los usuarios deben estar alerta al detectar comportamientos inesperados y poder reportarlos de inmediato al personal especializado.

Fuentes de información externas.-

Los virus informáticos son capaces de propagarse rápidamente debido a la forma en que la información fluye en este mundo moderno. Ese mismo flujo de información puede ser de gran ayuda para detectar una posible fuente de infección. Los periódicos, las revistas y grupos especializados han recopilado gran cantidad de información relacionada con los virus y otras formas malignas que atacan el software. Estas fuentes de información pueden ser valiosas para mantener un estado de alarma de las amenazas y las medidas disponibles para detectar o prevenir algunos virus específicos.

Recuperación de información.-

Una vez que el virus ha sido detectado e identificado, y la medidas se han tomado para evitar que se expanda, es necesario recuperar la información después de la infección. El principal objetivo de ésta actividad es proporcionar a cada usuario afectado un ambiente de computación libre de infecciones. Se debe asegurar que finalmente el ambiente quede totalmente desinfectado.

Estas actividades de recuperación se resumen en las siguientes :

- Remplazar todos los objetos infectados en el sistema con una versión desinfectada, y

- Restaurar cualquier otro objeto que la acción del virus pudo haber dañado.

Es de crítica importancia durante estas actividades evitar que se introduzca nuevamente el virus al sistema. Esto pudiera ser hecho, por ejemplo, restaurando un archivo ejecutable de una copia de respaldo.

Recomendaciones .-

Los virus informáticos son una amenaza para la seguridad de programas y datos en los sistemas de cómputo. En este trabajo se han dado algunas recomendaciones para evitar estas amenazas. Los puntos siguientes dan en forma resumida las acciones a tomar contra de los virus.

- Los virus representan relativamente una nueva amenaza para la seguridad de los sistemas de cómputo.
  - Se pueden propagar en forma automática, sin la intensión de la gente.
  - Se pueden propagar en forma en forma extensa y rápida dentro de una organización.
  - Pueden efectuar cualquier acción que el diseñador haya intentado.
- Los riesgos ocasionados por los virus pueden ser reducidos por las acciones adecuadas.
  - Seguir adecuados procedimientos de seguridad.
  - Educar a los usuarios sobre posibles amenazas de la información, incluyendo virus informáticos.

- Asegurarse de que se tengan respaldos adecuados de información importante.
- Establecer pasos para reducir la posibilidad de ser infectados.
- Aislar sistemas críticos de fuentes de infección, tales como redes de datos y programas externos.
- Limitar la capacidad de crear o instalar nuevos programas en aquellos sistemas que no requieren de esto.
- Asegurarse de que existe adecuado control en las bibliotecas de software, desarrollo de sistemas, y en las otras áreas similares de la organización. Esto incluye cambios en la administración y posiblemente el uso de programas que ayuden a la detección de virus.

- Tomar acciones para asegurar que las infecciones de virus se detecten en forma inmediata.
  - Educar a los usuarios para que conozcan posibles señales de alarma.
  - Usar programas que informen a los usuarios de cualquier modificación no intencional a programas y datos.
  - Asegurarse de que los usuarios sepan a quien dirigirse en caso de la amenaza de un virus.

- Tomar acciones para combatir las infecciones que sean detectadas.
  - Establecer un equipo de soporte que reaccione n forma inmediata ante un problema.

- Aislar sistemas infectados hasta que estos puedan ser eliminados para que se evite una posible infección a otros sistemas y que este pueda infectarse nuevamente.

- Finalmente, se debe estar preparado para recuperar la información después de que se ha detectado la presencia de un virus.

  - Establecer procedimientos de recuperación de información de respaldos no infectados.

  - Mantener especial cuidado en evitar un nuevo contagio del virus que se erradico después de que se hayan recuperado los respaldos.

SEGURIDAD EN REDES

Protección de redes de área local (LANs) .-

La protección de LANs (Local Area Networks) parece ser una tarea más difícil que la protección de microcomputadoras, debido a que el acceso a la información se hace posible en un área mayor. Esta dificultad adicional es, sin embargo, un mito, debido a que la información en LAN es típicamente manejada por una máquina llamada "server". En este caso los códigos de acceso y controles de acceso se vuelven más efectivos.

La intercepción de mensajes en LAN puede tener problemas significativos en redes de gran tamaño. Los analizadores de red pueden no ser usados para su propósito original de identificar y corregir problemas de la red, sino para interceptar información sensible. La necesidad de proteger una LAN contra ataques de este tipo se esta realizando gradualmente.

Las redes que abarcan grandes áreas (wide-area), junto con las formas más simples de transmisión de voz y datos, tienen diferentes problemas. En principio, se pueden usar muchos protocolos de seguridad elaborados para identificar la identidad de los participantes en un diálogo y para asegurar que la información no sea interceptada o alterada.

Tipos de protección para redes .-

El hablar de la protección en las redes es muy importante debido a que no se trata de protegerse a uno mismo, sino más bien, de la protección colectiva de los usuarios de una red de datos.

Se ha mencionado antes que la prevensión es la base para una adecuada protección de la información. El objetivo primordial de la protección en las redes es evitar el acceso a información confidencial.

Los intrusos que pretenden introducirse en una red han utilizado varios medios para violar la información. Estos medios van desde hurtar la información almacenada en disco flexible, hasta la introducción de virus informáticos que pueden desvirtuar lo que se ha almacenado en la red.

Realmente no existe una forma para proteger la información que se almacena en una red debido a que ésta depende de las necesidades de la empresa y de lo delicada que sea su información. Por ejemplo, si hablamos de los bancos, líneas aereas, compañías de seguros o secretarias de estado, la necesidad es obvia, pero en otros casos esto no es tan necesario. Un riesgo grande en una empresa sería que un empleado tenga acceso a la nómina, pronósticos de ventas, cuentas bancarias, etc.

En todo sistema de seguridad se puede tener protección de software y de hardaware. Es necesario evaluar en que casos aplica cada una o ambas.

Cuando se refiere a la seguridad del hardware se habla de la protección física del equipo de cómputo. Una forma de protección usando hardware son las estaciones de trabajo que no utilizan discos flexibles y que cargan el sistema operativo desde el servidor, sin embargo, se debe tener cuidado con las estaciones de trabajo que se conectan a tráves del puerto serial a algún modem para comunicaciones con algún otro sistema. Esta forma de acceso tiene una gran probabilidad de ser infectada por un virus. También, se debe de asegurar los cables, ya que como es en le caso de los de cobre, se puede tener radiación electromagnética la cual puede ser interceptada por equipos especiales. Una forma de evitar este problema es mediante el uso de fibra óptica.

Existen otros dispositivos llamados biométricos que permiten la identificación de una clave personal mediante el uso de huellas digitales o del muestreo de la retina del usuario. Obviamente éste tipo de sistemas son los más costosos cuando se habla de seguridad en la red.

Virus en las redes .-

La introducción de virus informáticos es un problema serio de seguridad. Una forma de protegerse de los virus informáticos es mediante el control de accesos.

Sin embargo, las dos características por las cuales se les considera funcionales a las computadoras personales son la capacidad de almacenamiento y procesamiento locales. Estas mismas características, son también la causa de los problemas de seguridad de las mismas.

Algunas compañias han tratado de combatir este problema evitando el uso de discos flexibles y con redes de área local (LANs). El evitar el uso de discos portables evita de alguna forma el plagio.

El manejo de redes locales puede ocasionar, por ejemplo, que un usuario autorizado obtenga información de la computadora central y la almacene en el servidor del LAN dejando de esta manera la información disponible a otros usuarios no autorizados. Este problema de seguridad generalmente es no intencionado, pero puede provocar grandes pérdidas.

Los respaldos de información presentan un gran riesgo para la seguridad de los datos. En estos casos, las redes de área local pueden ser de gran ayuda para efectuar los respaldos de archivos críticos que se actualizan frecuentemente y que residen en el servidor de la LAN.

Sin embargo, la introducción de nueva tecnología como los LANs, discos duros removibles, o almacenamiento óptico puede crear nuevos riesgos o regenerar otros que ya habían sido erradicados.

De esta forma, es importante evaluar los procedimientos de la organización como una alternativa para reducir los riesgos en la seguridad de la información.

ADMINISTRACION DE LA SEGURIDAD

La administración de los procedimientos de seguridad en empresas que manejan grandes volumenes de información requieren de un análisis adecuado de la situación de la empresa en cuanto información se refiere.

Como se ha mencionado antes, la seguridad puede afectar el trabajo de algunas personas. Esto debido principalmente al procedimiento burocrático que se establece para poder accesar, muchas veces en forma temporal, información crítica para fines de negocio.

En muchos casos un procedimiento de seguridad puede significar pérdida de dinero o simplemente un bajo aprovechamiento de los recursos humanos con que cuenta una empresa. Se han presentado casos en los que un procedimiento para accesar información, siguiendo las reglas de seguridad, toma varias horas cuando el trabajo productivo neto es solo de unos cuantos minutos.

Por otra parte, cuando no se controla el acceso indiscriminado a información sensible de la compañía, el daño provocado puede ser mayor. En este caso, se tiene que tomar una decisión tomando el mal menor. Es conveniente que se realice un análisis profundo de la situación para poder encontrar la forma adecuada de seguridad, o para simplemente evaluar la situación que cause un menor daño a la empresa.

Generalmente los sistemas de seguridad en empresas se basan en controles de acceso cuando son equipos grandes, y en protección física cuando son equipos pequeños.

El uso de passwords en las empresas para controlar el acceso es la forma más común en el uso de máquinas grandes. Los passwords se asignan a personas que necesitan accesar algún tipo de información con la previa autorización del responsable del sistema.

El área de seguridad es la encargada de proporcionar accesos a ciertas areas, con ciertos privilegios, además de cancelarlos por mal uso o por accesar areas de información no autorizadas.

El controlar los accesos no es solo para impedir el acceso a cierto tipo de información; los accesos se controlan para evitar que el trabajo que se realice en un area de pruebas pueda afectar la información de un área productiva.

Las normas de seguridad para equipos pequeños varián de acuerdo, también, a las necesidades de cada empresa. Por ejemplo, no son las mismas necesidades de seguridad para una empresa que tenga computadoras para procesar exclusivamente su nomina que otra que se dedique a comercializar software. En cada situación las necesidades de protección varian de acuerdo a la importancia de nuestra información hacia la empresa.

En el caso de la compañía que comercializa software se puede incurrir en el plagio, posiblemente en la etapa de desarrollo, lo cual repercutiría en el mercado en el cual se comercializará dicho producto. Para la empresa, este problema puede ser individual con repercusiones de tipo interno y pérdidas ecónomicas menores.

En el caso de equipos pequeños la seguridad puede ser más "sencilla"; quizá el guardar el equipo de computo bajo llave sea suficiente.

Cuando se maneje una computadora personal, es importante que el usuario maneje ciertas reglas básicas para el uso y cuidado del equipo; también, es importante que el grupo encargado de la seguridad impida el uso de software externo de dudosa procedencia, y organice campañas de vacunación anti-virus.

2 CRIPTOGRAFIA

La criptografía es considerada una de las formas más eficientes de codificar la información en un lenguaje "secreto" para la transmisión de información. La eficiencia en la codificación de un código de encripción depende básicamente del algoritmo que se utilice para encriptar el mensaje o archivo que se quiere proteger. Cuando se utiliza un código secreto de encripción, el emisor y el receptor deben tener una misma clave que les ayude a codificar ó decodificar el mensaje.

Existen varias formas de encripción, muchas de las cuales basan su eficiencia y confiabilidad en la complejidad de su algoritmo. También se considera que el mantener en secreto éste algoritmo de encripción puede dar resultados confiables, pero esto no necesariamente es cierto y podemos hacer uso de esta circunstancia para elaborar un sistema de seguridad confiable. Estos aspectos se explican en éste capítulo.

ENCRIPCION DE DATOS

Antecedentes.-

La forma más fácil de instrumentar la encripción en una microcomputadora es con un programa de encripción y desencripción. Con un comando, se encriptará y se desencriptará un archivo específico con un código de acceso específico. Tales programas son simples de escribir pero poco prácticos en su uso. Por ejemplo, para editar un documento previamente encriptado durante una sesión típica con un procesador de palabras se deben seguir los siguientes pasos :

- salir del procesador de palabras;
- llamar al programa desencriptador, especificar el nombre del archivo con la llave apropiada de encripción, y esperar a que todo el archivo sea desencriptado;
- volver a entrar al procesador de palabra;
- editar el documento;
- salir del procesador de palabras;
- llamar al programa de encripción, especificar el nombre del archivo, la llave apropiada de encripción, esperar a que todo el archivo sea encriptado, y
- volver a entrar al procesador de palabras.

Además se deben borrar los archivos de respaldos temporales que pudo haber generado el procesador de palabras. Si éste creó o borró cualquier archivo temporal por si mismo, se debe borrar el espacio libre en disco en el caso de que alguna información sensible se hubiera almacenado. Un refinamiento de dicho método es hacer los programas de encripción residentes en memoria y hacer que puedan ser activados por una simple tecla. Esto desplaza la "clave de salida" y la "clave de entrada" del procesador de palabras, pero por otro lado, hace que el método se vuelva vulnerable.

Debido a la impaciencia humana puede ocurrir que un documento se deje sin encriptar y que los usuarios encuentren el procedimiento más "comodo" y de esta forma eviten la pérdida de tiempo.

Una de las principales debilidades de un sistema de protección basado en la encripción es el uso de llaves. Si se tiene que introducir llaves de encripción y desencripción frecuentemente, seguramente mucha gente seleccionará llaves de acceso cortas, y por tanto inseguras, con el fin de acelerar el procedimiento de protección. Además no existe protección interna cuando se escribe mal una clave. Suponiendo, que la llave CONFIDENCIAL, se escribe erroneamente como COMFIDENCIAL, todo parecería estar bien hasta que se trate de accesar el documento quizá meses más tarde, y encontrar que todo es ilegible.

La encripción transparente soluciona estos problemas. Un sistema de encripción transparente se basa en un programa TSR (" Terminate and Stay Resident ") que permanece permanentemente activo. Este monitorea e intercepta todos los accesos a disco para archivos sensibles. Cuando los registros son escritos, el sistema de encripción los encripta antes de que se transfieran al disco y cuando los registros son leidos, son desencriptados antes de que sean manejados por el paquete o la aplicación de software que este usando el usuario en esos momentos.

Un sistema de encripción transparente es invisible al programa de aplicación. Esto elimina problemas de compatibilidad. También , la protección se extiende a los archivos temporales que el programa de aplicación puede crear y borrar sin el conocimiento directo del usuario.

La encripción transparente es invisible para el usuario. Las llaves de encripción se deben dar solo una vez al comienzo de cada sesion y permanecen durante el desarrollo de la misma (adecuadamente encriptadas y ocultas) en la memoria privada del programa TSR. Por tanto, después de dar al inicio de la sesion el "log-in", la secuencia compleja de operaciones necesaria para editar un documento confidencial con un programa simple de encripción se convierte en simplemente "editar el documento".

Debido a que sólo se introducen las llaves una sola vez, hay menos resistencia por parte del usuario hacia su uso, y de esta forma se soluciona el problema tradicional de escoger códigos de acceso cortos, memorizables y faciles de adivinar mediante la restricción mínima de un código de acceso.

Con la encripción transparente el problema de introducir una llave erronea no tiene efectos catastroficos en lo que ya esta almacenado. Todo lo que sucede es que todo lo que se lea del disco con la llave incorrecta aparecerá ilegible : una vez que se introduce la llave correcta, todo es legible nuevamente.

Una última ventaja que se encuentra en la encripción transparente es su eficiencia en las aplicaciones de base de datos. En lugar de tener que desencriptar y luego reencriptar toda una base de datos, posiblemente con algunos megabytes de tamaño, cuando se requiera un registro, un sistema de encripción transparente desencripta sólo la información que realmente se necesita, algunos apuntadores, y el registro requerido. El resultado que se obtiene en rapidez hace la diferencia en términos de funcionalidad.

Las características que se deben de buscar en los sistemas de encripción transparente son velocidad, granularidad, transferibilidad y seguridad. La velocidad es una necesidad básica; si la encripción o la desencripción tardaran en su ejecución, el tiempo de respuesta sería grande.

La granularidad se refiere al grado de detalle que se puede especificar para las llaves de encripción. Los sistemas de encripción que interactuan a nivel usuario pueden manejar un buen nivel de granularidad, con una llave diferente para cada archivo si es necesario; los sistemas de encripción transparente tienen más dificultades para manejar la granuliridad, ésto debido a que se tiene poca interacción con el usuario.

La transferibilidad muchas veces es pasada por alto ya que siempre se tendrá la necesidad de intercambiar los archivos entre computadoras. Esto quiere decir, que se debe evitar el uso de llaves diferentes, y que todo mundo use la misma llave para que de esta manera exista una adecuada tranferencia de información.

La seguridad es la característica más difícil de evaluar de todas. Muchos proveedores de sistemas de seguridad subestiman las sutilezas, escollos y peligros de la criptografía y de algunos productos actuales que son inseguros, muchos de los cuales pueden ser violados en pocas horas y con un mínimo de conocimiento acerca de criptografía.

Desafortunadamente, el algoritmo de encripción parece ser el último aspecto que se considera para diseñar sistemas de seguridad. Al observar algunos productos comercialmente disponibles, se puede encontrar el uso de algoritmos triviales y muchos de estos usando generadores de números aleatorios, ó "cihpers" tipo "stream", los cuales se diseñan para usarse una sola vez en transmisión de datos y, para encriptar en forma monotona bloques de datos.

Lo más peligroso acerca de tales errores es que son completamente invisibles al usuario que no tiene experiencia alguna en criptografía. Esto se puede detectar si se manda a imprimir el texto ilegible o si la base de datos carga información erronea. Pero en un sistema de seguridad, todo lo que se puede hacer es checar que los archivos sean legibles después de ser encriptados o desencriptados. Más aún, sólo se puede esperar que el sistema sea realmente seguro y consistente.

Se puede asegurar que el algoritmo de encripción es "solo parte" del sistema de seguridad completo. De hecho lo es, pero, es una parte indispensable. Los castillos de un edificio son sólo parte de la construcción total, pero sin ellos el edificio seguro se vendría abajo; lo mismo sucede con los sistemas de encripción.

ESTANDARES DE ENCRIPCION

Tradicionalmente en las áreas de funcionalidad donde se encuentran problemas para formular juicios propios se encuentran los estandares, y lógicamente se puede esperar que suceda lo mismo con la encripción.

En los Estados Unidos existe el "Data Encryption Standard" (DES), con una serie de variantes en relación a su uso. Desafortunadamente, el DES es un estándar para "hardware";y decimos desafortunadamente, por que esto significa adicionar más hardware a nuestro equipo de cómputo reduciendo en esta forma sus recursos y capacidades.

El estándar DES se basa en la aplicación de la manipulación de bits y por tanto, hace que al aplicarse en "software" sea muy lento. El estándar DES es un "cifrador" de flujo con un tamaño de bloque pequeño (8 bytes), orientado más a la transmisión de datos que al almacenamiento de información, así que no es recomendable, ni prudente usarlo para un sistema de encripción transparente.

El sistema DES tiene diez años de existir, pero es poco probable que sea desplazado por otro estándar, aún a pesar de que este más orientado al almacenamiento que a la tranmisión de datos.

Los estándares tienen como gran ventaja que consolidan el progreso, pero tienen como gran desventaja que impiden el desarrollo a futuro.

Cuando se diseño originalmente el DES, había una gran necesidad por la estandarización en "hardware", debido al tamaño de la inversión requerida para diseñar circuitos integrados orientados a la encripción y a que la comunicación de datos necesitaba el mismo algoritmo en ambos extremos del enlace (emisor-receptor).

Pero hablando de "software", principalmente sobre diseño para una operación local en una microcomputadora simple, la inversión es menor y existe menos necesidad de uniformidad y de esta forma el gasto e inconveniencia de un estándar no es justificable.

Se ha sugerido también, la creación de un cuerpo rector cuya función sea evaluar los algoritmos de encripción y reportar su eficiencia en cuanto a seguridad se refiere. Desafortunadamente, esto es imposible. Primero, la seguridad no es materia de comprobación (excepto para aquellos algoritmos obviamente inseguros), y segundo, la experiencia y opinión debe tomar parte en la evaluación.

SISTEMAS DE ENCRIPCION CONVENCIONALES

Un método importante de encripción es el de sustitución. Este método consiste en remplazar cada ocurrencia de una letra o una palabra (ó byte), con otra letra o palabra. El operador XOR es una forma conveniente para instrumentar el método de sustitución con computadoras. Cuando se aplica un XOR a dos bits, el resultado es '1' si y sólo si uno de los bits de entrada es '1'. El resultado es '0' si ambos bits son '0' ó '1'.

La función XOR es conveniente debido a que es rápida y se pueden desencriptar archivos simplemente aplicando un XOR al "ciphertext" con los mismos datos que se usaron para encripatar el código fuente o "plaintext". Por ejemplo, se puede encriptar la palabra HOLA aplicando un XOR en cada byte con la representación ASCII de la letra A (0100 0001). Para encriptar el "plaintext" se usa la llave "A" y se aplica un XOR; para desencriptar, se usa la misma llave "A" y se aplica un XOR.

(a)

HOLA	0100 1000	0100 1111	0100 1100	0100 0001
LLAVE (letra A)	0100 0001	0100 0001	0100 0001	0100 0001
CIPHERTEXT	0000 1001	0000 1110	0000 1101	0000 0000

(b)

CIPHERTEXT 0000 1001 0000 1110 0000 1101 0000 0000

LLAVE
(letra A) 0100 0001 0100 0001 0100 0001 0100 0001

PLAINTEXT 0100 1000 0100 1111 0100 1100 0100 0001
(ciphertext XOR llave)

La eficiencia de un buen sistema de encripción no depende de mantener su algoritmo en secreto; el éxito del código encriptado (ciphertext) depende del secreto de su llave.

El análisis estadístico puede descifrar un sistema de encripción simple siguiendo ciertos patrones del lenguaje común. El lenguaje natural tiene ciertos patrones tales como, la frecuencia de uso de las letras, la combinación común de las letras y la frecuencia de uso de algunas palabras, entre otras.

Esta serie de patrones, que aparecen en el "plaintext", apareceran también en el "ciphertext" aunque su expresión sea diferente. Una vez que se conozcan estos patrones, estos pueden ser usados para romper el "ciphertext" e identificar el código encriptado.

En forma alternativa, se puede romper el código de encripción mediante un ataque de "fuerza bruta" tratando las 256 opciones posibles (del 0000 0000 al 1111 1111), haciendo el barrido con el uso de una computadora en pocos segundos.

Una manera de solucionar este inconveniente es usando llaves más largas. Por ejemplo, se podría usar una llave de 4 letras como A5GE (una llave aleatoria exelente). En tal caso, se encripta el primer byte con A, el segundo con 5, el tercero con G, y el cuarto con E.

Después de barrer la encripción con las 4 letras estas se reusan; así, la encripción del quinto byte se hace usando la letra A nuevamente y así las demás letras. El tamaño de la llave es de 4 y esto hace que sea más difícil descifrar el código usando métodos de seguimiento de patrones en las letras para romper el mensaje; claro que es más difícil descifrar el código, pero no imposible.

Desafortunadamente, si los decifradores de códigos saben (o pueden adivinar) parte del "plaintext" (e.g., si saben que cualquier mensaje comienza con "hola amigo"), entonces, se pueden usar análisis criptografía analíticos para decifrar la llave. En el ejemplo anterior, se puede ver como aplicando un XOR al "plaintext" con el "ciphertext" se decifra la llave.

Idealmente, se debe tener una llave que no se repita. En este caso se debe usar una llave compuesta de bits aleatorios que no se reuse, la cual es llamada cinta de un tiempo.

Se puede probar matemáticamente que para un sistema de encripción basado en una cinta de un tiempo, es imposible el descifrado.

Desafortunadamente, Una cinta de un tiempo requiere de una llave tan larga como el mensaje que se requiere encriptar; así que se tiene el problema de transmitir el mensaje en forma segura. Puede parecer al principio, que se crea un sistema de criptografía de cinta de un tiempo, extendiendo una llave corta con un generador de funciones de computadora, usando la llave corta como semilla.

Aunque muchos paquetes comerciales de encripción usan generadores de funciones, se deben considerar más estos como una diversión que como un sistema de encripción.

Un generador de números aleatorios de computadora, genera realmente números seudoaleatorios. Existe relación matemática entre un número generado y el otro que le sigue. Consecuentemente, tales sistemas de encripción, frecuentemente descritos como "inviolables", pueden ser descifrados generalmente en minutos.

Para evitar diferencias entre los sistemas de encripción la NBS (National Bureau of Standars) estableció el DES (Data Encryption Standard). El estándar DES fue originalmente desarrollado por la IBM y adoptado como estándar por la ANSI.

Antes de ser designado como estándar, el DES fué validado por la NSA (National Security Agency), la cual encontro que el estándar estaba protegido contra cualquier herramienta estadística o matemática. Desde su adopción como estándar, DES ha sido usado por la mayoría de los bancos y agencias del gobierno en los Estados Unidos (exepto por la milicia).

El DES, trabaja con un bloque de 8-bytes (64 bits). El proceso de encripción es controlado por una llave de 54 bits, esto es, 72,057,594,037,927,946 llaves posibles.

Cada bit de salida es una función compleja de cada bit del bloque de entrada y de cada bit de la llave. La descripción aplicando DES se hace en forma reversible aplicando el algoritmo en forma inversa. El proceso de encripción consiste de una permutación inicial del bloque de entrada seguido por 16 rondas de cifrado, y finalmente una inversión de la permutación inicial.

Después de la permutación inicial, el bloque que esta siendo encriptado se divide en dos partes, llamadas **Lo** y **Ro**. En cada una de las 16 rondas de cifrado, la parte **L** nueva, es la ronda previa de la parte **R**. La nueva parte **R** es la ronda previa de la parte **L** aplicada con un **XOR** al resultado de la función cipher. Esto es, la salida de la ronda **i** es :

Li = Ri - 1

Ri = Li - 1 XOR f(Ri-1, Ki)

La función "cipher" deriva su salida basada en la parte **R** anterior **(Ri-1)** y la llave de la ronda actual **Ki**. Se usan las entradas para ejecutar una sustitución por medio de ocho tablas llamadas cajas "**S**" y todas las permutaciones requeridas.

La eficacia de DES ha sido comprobada por la NSA a través del análisis y amplio uso por años sin haber tenido alguna violación al sistema. La gran debilidad de DES es su longitud de llave limitada. Se critica que DES puede ser violado con un ataque de "fuerza bruta" (i.e., tratando todas las llaves posibles). Sin embargo, tratar todas las llaves posibles dentro de un tiempo razonable requeriría de una máquina especial que usaría un millón de procesadores trabajando al mismo tiempo en forma concurrente. Cada procesador desencriptaría el texto cifrado (ciphertext) usando un conjunto de llaves diferentes y checaría (e.g., usando un diccionario) si se adivino la llave correcta. Más aún, costaría millones de dolares construir tal máquina. Sin embargo, estos defectos no son una razón suficiente para no usar DES. Si se tiene preocupación por detalles como este, se puede aún más complicar el proceso de encripción de DES mediante el uso de una etapa adicional previa a la encripción.

SISTEMAS DE ENCRIPCION DE LLAVES PUBLICAS

Cuando se usa un sistema de encripción convencional como DES, tanto el emisor como el receptor, deben de conocer la llave de encripción (y desencripción) con que se encripto la información. Por tanto, se necesitan medios seguros para transmitir la llave de uno a otro lado. Si se cambian las llaves frecuentemente, la transmisión de estas se vuelve un problema mayor. Mas aún, con un sistema de encripción convencional es imposible comunicarse con alguien nuevo antes de que se hayan intercambiado en forma segura las llaves; ésto puede llevar mucho tiempo. Los sistemas de encripción de llaves públicas fueron diseñados para pasar por alto estos problemas.

Los sistemas de encripción de llave pública están basados en el uso de una función con una puerta secreta de un solo sentido. Se puede usar ésta función para encriptar la información en un solo sentido. Para aplicar la función en la dirección contraria, usada para desencriptar la información, se debe tener cierta información secreta; de ahí el nombre de puerta secreta.

En un sistema de encripción de llave pública, cada persona tiene dos llaves: una para encriptar, **Ea**, y otra para desencriptar, **Da**. Al desencritar con **Da** un "Plaintext" **P** que fué encriptado usando **Ea** restaura el "Plantext" original, ésto es, **Da(Ea(P)) = P**. Tanto **Ea** como **Da** deberían ser fáciles de procesar, pero conociendo **Ea** no posible descubrir **Ea**.

Si se usa un sistema de encripción de llave pública, se puede publicar la llave de encripción **E** (la llave pública) en un directorio público, mientras que la llave **D** (la llave privada), se mantiene en secreto. Si alguien desea transmitir un mensaje, todo lo que la persona tiene que realizar es buscar la llave pública **(Ea)** y usarla para encriptar el mensaje como **Ea(P)**. Solamente la persona que conozca la llave privada **Da**, podrá desencriptar el mensaje al "Plaintext" original, **Da(Ea(P)) = P**.

Sistema de encripción RSA .-

El sistema de encripción de llave pública más importante que existe en la actualidad es el RSA, llamado así en honor a sus inventores, Rivest, Shamir, y Adlemar. Para usar el RSA, se necesita escoger, en forma aleatoria, dos números primos grandes, llamados **P** y **q**. Luego obtener **n** como el producto de los dos primos: **n = p q**. Luego escoger aleatoriamente, un número grande, tal que **d** sea relativamente primo a **(P-1) * (q-1)**; en otras palabras, el máximo común divisor de **d** y **(P-1) * (q-1)**; es **1**. Finalmente, obtener **e** tal que **(e*d)** módulo **((P-1) * (q-1)) = 1**. La notación "**X módulo Y**" significa que se obtendrá el residuo de dividir **X** por **Y** usando una división entera. Por ejemplo, 2 módulo 5 = 0, puesto que 20/5 = 4 con residuo 0; 13 módulo 3 = 1 puesto que 13/3 = 4 con residuo 1.

La llave pública es el par de números **(e,n)**, y la llave privada es **(d,n)**. Aunque **n** y **e** son públicos, es difícil de llegar a **d**, puesto que no existe un algoritmo eficiente para factorizar números grandes. Consecuentemente, para asegurarse, **p** y **q** deben ser muy largos (al menos cien dígitos numéricos), para que **n** sea extremadamente grande (al menos doscientos dígitos) y que no puede ser factorizado dentro de un tieMpo raronL)Seîb é!Ç"Ç'"3ôß#ôôcÿ RSA, primero se divide el "Plaintext" en bloques que pueden ser representados como un entero entre **0** y **n-1**. Entonces, se encripta cada bloque elevándolo a la potencia **e** módulo **n**. Para desencriptar el bloque, se eleva a la potencia **d**, módulo **n**; esto es, **C = pe módulo n**,y **P = Cd módulo n**.

A continuación se muestra un ejemplo de como usar RSA. Para simplificar el ejemplo, se usan primos pequeños para **p** y **q**.Sin embargo, para crear un sistema seguro, se deben de usar números muy grandes.

- Supóngase que se escoge **p = 3** y **q = 11**.

- Luego, **n = pkq = 3*11 = 33** y
  (P-1)*(q-1) = 2*10 = 20

- Se puede usar **d = 7**, puesto que 7 es primo relativamente a 20.

- Luego se necesita encontrar un e,tal que **e*7 módulo 20 = 1**.

- Entonces, la llave pública es **(3,33)** y la privada será **(7,33)**.

Si se representa el mensaje usando un **1** para **A**, un **2** para **B**, un **3** para **C**, y asi consecutivamente, el "Plaintext" **CADA** se escribiría como **3/4/**. La tabla siguiente muestra como encriptar ésta palabra usando la llave pública **(3,33)**.

P	Pe	Pe módulo n
3	27	27
1	1	1
4	64	31
1	1	1

Entonces, el "Cihpertext" sería **27 1 31 1**.

Para desencriptarlo, se hará de la siguiente forma para restaurar el "plaintext" original.

e	Cd	Cd módulo n
27		3
1	1	1
31	27512614111	4
1	1	1

El algoritmo del RSA se conoce desde 1978, y no se sabe de algún caso en el que éste haya sido violado; su fortaleza se basa en la complejidad para factorizar números muy largos, por lo cual alctualmente se cree que no exista algún algoritmo descifrador para el RSA. Si tal algoritmo se encuentra, el algoritmo RSA sería desechado. Más aún, nadie ha probado que factorizar **n** es esencial para conocer la llave privada.

De una manera más práctica, La operación del RSA con números muy grandes hacen el sistema demasiado lento. Además, el algoritmo RSA está patentado y por lo tanto no puede ser usado en forma indiscriminada.

FIRMAS DIGITALES

Además de asegurar la privacidad, la encripción puede ser usada para verificar la autenticidad. Supongase que se le manda un mensaje al corredor de bolsa para que venda todas las acciones. ¿Como puede el corredor de bolsa verificar que se envió tal mensaje? Si se tiene duda aún siendo la persona autorizada, como puede verificar el corredor de bolsa que realmente fué la persona indicada.

Si se hiciera por medio de correo postal, la firma de la persona autorizada sería la forma de verificar la autenticidad, ¿ pero como se hace para mensajes electrónicos?

El encriptar el mensaje usando una llave conocida solo para la persona que envía el mensaje el corredor de bolsa no resuelve el problema.

Los sistemas de llave pública proporcionan una solución simple y elegante creando firmas digitales.

Una compuerta secreta de una sola dirección tiene la propiedad de **D(E(P)) = P**. Si la función usada por el sistema de encripción de llave pública también tiene la propiedad de **E(D(P)) = P**, se dice que es una permutación de compuerta secreta.

El sistema de encripción de llave pública RSA, cuenta con éste requisito. Usando un sistema de llave pública se puede encriptar el mensaje usando la llave privada **Da**.

Cualquiera que reciba el mensaje **Da(P)** puede desencriptarlo usando la llave pública Ea. ya que **Ea(Da(P)) = P**.

Debido a que la llave **Da** es conocida sólo por el transmisor, el receptor sabe y puede probar, quién es el autor del mensaje.

Si se quiere enviar un mensaje privado que pueda ser verificado por alguien más, entonces se debe encriptar **Da(P)** con la llave pública de esa persona, quedando **Eb(Da(P))**.

Si se usa la llave privada, **Db**, esa persona obtendrá **Db(Eb(Da(P))) = Da(P)**, la cual será guardada como prueba de autenticidad y luego desencriptar **Da(P)** usando **Ea(Da(P)) = P**. Por tanto, la privacidad y la autenticidad se llevan a cabo.

3 DISEÑO DEL PROTECTOR

El diseño se divide en dos partes fundamentales, una parte de software y la otra de hardware. Ambas juegan un papel importante en la protección de la información. Los protectores que usan sólo software como medio de protección son generalmente insuficientes y necesitan un complemento para que puedan ser efectivos. La protección con hardware es recomendable debido a que los circuitos eléctronicos son dificiles de violar y existe un mayor desconocimiento del funcionamiento de los mismos por parte de los usuarios.

En el proyecto se esta considerando un programa en ensamblador que trabaje residente en memoria y que además tenga la capacidad de comunicarse a través del puerto paralelo con un circuito protector externo. Dicho protector tendra la capacidad de almacenar información encriptada y accesarla y cada vez que se accese información al disco flexible (ó duro) desde cualquier aplicación que haga uso del BIOS para accesar el disco del sistema.

El programa principal que permanecerá residente en memoria detectará cuando exista una solicitud para acceso al controlador del disco y manejará la interrupción 13h del BIOS en forma distinta a un acceso normal.

Se detectará cuando se tenga una solicitud de lectura o escritura y la interrupcion normal del BIOS será cambiada por una rutina que tiene como funciones principales, comunicarse con el circuito externo y realizar la encripción de la información que se almacene en el buffer de lectura que se asigna durante un acceso de lectura/escritura.

El circuito externo tiene como principal función el recibir la información proporcionada por el programa residente, manejarla y encriptarla. Aquí es necesario mencionar que el programa en software pude tener un programa encriptador o parte del mismo que se maneje con el circuito protector. El circuito protector puede recibir o mandar información relacionada con el dato mismo o pueden ser instrucciones o información para encriptar.

Los circuitos del tipo Bit-Slice pueden proporcionar muchas opciones de diseño debido a la flexibilidad que estos tienen para programar microcódigo. Estos circuitos pueden ser microprocesadores muy poderosos que se pueden adaptar a las necesidades del instrumentador.

De acuerdo al nivel de especialización que tengan las instrucciones de un microprocesador pueden ser del tipo RISC (de Reduced instruction set computer) o tipo CISC (de Complex instruction set computer). Es importante mencionar que cuando se requiere aplicar un microprocesador a una tarea muy especifica, es conveniente el uso de un microprocesador tipo CISC ya que el grupo de instrucciones del microprocesador es muy particular y la instrucciones de programa tiene más ciclos de reloj.

Las aplicaciones que se pueden dar a los de microprocesadores bit-slice son multiples y dependen de los requerimientos de cada proyecto. Este tipo de diseños se pueden utilizar como sustitutos de microprocesadores no dedicados y que requieren de cierta especialización.

Por ejemplo, se puede hablar del caso de un adquisidor de datos para procesar información de tipo climatológica, el cual tiene como procesador central un Z80, 8086, 80286,etc, sin embargo, el uso de un microprocesador de este tipo puede no ser eficiente debido, principalmente a que un procesador de los antes mencionados no cuenta con las instrucciones de microprograma que puedan cubrir nuestras necesidades de procesamiento.

En pocas palabras, si nuestro sistema de adquisición de datos tiene que trabajar en tiempo real recolectando datos de información pluvial, temperatura ambiente y velocidad de viento en un mismo tiempo, además de proporcionar la información en forma inmediata (interacción hombre-máquina) sin la necesidad de hacer un barrido sobre la memoria RAM del adquisidor y sin dejar esclavizada la computadora personal a un sistema de adquisición de datos, lo más seguro es que el microprocesador comercial nos pueda ayudar si ha éste le añadimos un poco de diseño electrónico.

Si optamos por usar un secuenciador microprogramado podemos diseñar un circuito dedicado que pueda recolectar datos y al mismo tiempo, si usamos conjuntamente un programa residente en memoria que se encargue de vaciar la información en el disco de la PC, se puede transferir la información a la PC sin necesidad de dejarla esclavizada, además de poder accesar la información muestreada en forma "inmediata".

El uso de este tipo de circuitos en el diseño de adquisidores de datos nos permite hacer un uso eficiente del equipo de cómputo que hace la recolección de los datos, ya que estará operando en forma interactiva con el usuario al mismo tiempo de que se encuentre recibiendo información de los sensores, y transmitiendo información a través del puerto serial o paralelo.

En este capítulo se explicará cada una de las etapas que componen el diseño del protector, tanto en software como en hardware.

Otra aplicación que se puede dar a este tipo de sistemas puede ser el manejo de servomecanismos industriales controlados desde la PC. Usando estos sistemas, a través de un acoplamiento con motores analógicos o de pasos, se pueden utilizar para programar robots industriales con varios grados de libertad.

El diseño de un servomecanismo incluye el manejo de los motores de pasos a través de la PC para proporcionarle el número de pasos que debe girar el motor; para el caso de motores analógicos, se incluiría en el diseño un convertidor digital/analógico.

En el caso de motores analógicos la inercia puede ser un problema para controlar el número de grados que este debe girar. Se recomienda utilizar motores de paso en las aplicaciones en las que se necesite un grado muy alto de exactitud con poca fuerza de inercia, y utilizar motores analógicos para aplicaciones donde la inercia a vencer sea mayor y no se requiera de una gran precisión. Obviamente, si se opta por el uso de un motor analógico o de cualquier otro dispositivo que consuma demasiada corriente, se debera tener en cuenta el diseño de una etapa de potencia.

DISEÑO DEL SOFTWARE

El software forma parte esencial del manejo del protector. Los conceptos más importantes que fundamentan la programación del protector son la programación de rutinas que permanecen residentes en memoria, el manejo de las interrupciones en el procesador y, un conocimiento adecuado del dispositivo que se manejará; en este caso, los discos de almacenamiento secundario. El conocimiento que se necesita sobre los discos de almacenamiento secundario no sólo se refiere a su operación, es necesario tener un conocimiento adecuado de la forma en que opera el manejador de discos flexibles.

El programa que controla el circuito externo de protección se controla a través de un programa residente en memoria que detecta cualquier tipo de acceso a un disco flexible o al disco duro del sistema; después, mediante el uso del buffer, el cual se asigna por el manejador de discos, se encripta la información que se va a escribir o a leer.

El resultado, es un programa capas de detectar cualquier acceso a disco y encriptar la información mediante el uso de un circuito externo. El programa se carga una sola vez a través del AUTOEXEC o de forma directa.

En ésta parte se explican los conceptos que se utilizan para el diseño del protector que se relacionan con la parte de programación.

PROGRAMAS RESIDENTES

La parte medular del protector de software, en lo que se refiere a la interface de la parte de hardware con la PC, se basa principalmente en la programación con residencia en memoria. A éste tipo de programas se les llama TSR tienen muchas diferencias con los programas "normales" (procesadores de texto, hojas de cálculo y otros). Una de las diferencias más obvias es que después de que "terminan" su ejecución se mantienen almacenados en memoria sin que sus recursos se reasignen o liberen.

Un programa TSR ó residente en memoria, puede tomar el control de otro programa en diferentes formas; la más importante se realiza a través de una interrupción.

Originalmente los programas residentes se veían como una extensión del sistema operativo y no tenían el propósito de ser una función para crear programas de utilería como el "sidekick" y otras.

Existen dos tipos de programas residentes, los activos y los inactivos. Acontinuación se describen estos dos tipos de programas residentes.

Programas residentes activos.-

Este tipo de programas son los más comúnes y un ejemplo de este tipo de programas residentes es el "sidekick". Cuando una utilería de este tipo se activa, con una tecla "latente", toma el control de la computadora y ejecutan la función antes de que regresen el control al programa que llamó.

En el diseño del protector se descarto la opción de utilizar un programa residente activo,ya que como es obvio se requiere que la interacción con el usuario sea mínima.

Programas residentes inactivos.-

Este tipo de programas responden cuando de un determinado programa se hace referencia a una interrupción. Cuando se llama la interrupción, se ejecuta una función definida, similar a una subrutina, y luego regresan el control al programa que originalmente llamó a la interrupción.

Este tipo de programa es el que se usará para la construcción del protector de software. Esto debido, a que el protector, deberá actuar cuando se requiera un acceso a disco a tráves de una interrupción al procesador en forma transparente al usuario. Esto se explicará con más detalle en los siguientes parrafos.

Existe un problema al que se enfrenta con el sistema operativo DOS. El problema con el DOS es que no es reentrante (no se puede romper a la mitad una rutina interna del DOS y comenzar nuevamente). Si el DOS se encuentra procesando algo (un acceso a disco, por ejemplo) cuando un programa residente se activa para escribir algo en disco, se tendrán serios problemas y posiblemente se pierda la información de todo el disco.

Se han tenido algún progreso muy significativo con los programas residentes desde que se introdujo el DOS. Sin embargo, estos no se han llegado a difundir.

La función original para programar rutinas residentes (int 27h) ha sido reemplazada por la int 21h, con función 31h. Esta función es más conveniente de usar debido a que permite que se regrese un código de retorno y además de que permite usar más de 64k de memoria. Ambos factores justifican el uso preferente de la int 21h con función 31h para dejar rutinas residentes.

Cuando se ejecuta un programa residente, se establecen sus tablas de memoria y se prepara para ejecutarse mediante la interface con una interrupción del DOS. Cuando todo está listo, el programa debe determinar cuanta memoria se necesita mantener; después, se asigna 31h al registro AH, el código de retorno a AL y el número de párrafos a ser alojados (bloques de 16 bytes) para el programa residente a DX. Cuando el programa termina, la cantidad de memoria disponible para la ejecución de los programas se reduce en la cantidad asignada al programa residente, y el código de salida se regresa al "padre" (programa que llamó originalmente al programa residente).

Existen programas residentes muy simples, pero para aplicaciones más complejas existen varios detalles que se deben considerar.

Primero, se debe tener la forma en que se disparará el programa residente. Se puede pensar en amarrar el programa residente a la interrupción del reloj y activar el programa cada determinado número de segundos.

Aún se puede ligar el programa residente a la interrupción de teclado con una tecla específica. En estos casos se tienen que cuidar las colisiones debido a que una tecla puede ser usada en la aplicación que funciona como "padre".

Cuando se utilizan funciones del DOS, los programas residentes y las rutinas de servicio de interrupciones (ISRs) corren un mismo peligro.

El sistema operativo MS-DOS fué diseñado para servir a un sólo usuario, a una sola tarea y por tanto como se mencionó antes, no es reentrante.

Debido a que no existe una sincronización entre las funciones del DOS, los programas residentes ó las rutinas de servicio de interrupciones toman el control de la computadora cuando se encuentra ejecutándose una función del DOS.

Realmente el instrumentar un programa residente eficiente es una tarea de gran análisis para incluir características especiales en dichos programas.

MANEJO DE INTERRUPCIONES

Las interrupciones han existido por años. Las interrupciones no tienen una reputación agradable,y esto es debido a que se introdujeron por primera vez como la parte medular del diseño de sistemas y no dejan de ser una parte obscura y poco agradable para su aprendizaje.

En los sistemas de computación anteriores, las interrupciones eran los dolores de cabeza de muchos programadores sin experiencia.

Las interrupciones no han sido del dominio de los programadores de sistemas, si no más bien de los ingenieros de Hardware de tal forma que muchos programadores temen usarlas debido principalmente a la falta de conocimientos sobre el funcionamiento del hardware en una computadora.

El manejo de interrupciones en las computadoras personales pueden ser benigno si no se siguen las indicaciones correctas para su manejo. Si, se siguen las indicaciones generales acerca del manejo de las interrupciones se pueden programar fácilmente rutinas complejas. Sin embargo, cuando se ha obtenido la suficiente experiencia escribiendo rutinas que manejen interrupciones, esto se vuleve una tarea sencilla.

TIPOS DE INTERRUPCION

Una interrupción es una señal que va al procesador indicándole que un evento que requiere atención especial se ha llevado a cabo.

Si no se tuviera una señal de interrupción, se tendría que estar sondeando si un determinado evento ha ocurrido. Esto, obviamente da como resultado un desperdicio de los recursos.

Las interrupciones se pueden dar en forma externa, por dispositivos como el controlador de disco, o por eventos internos, tal como una división por cero en el cálculo en un programa.

Siempre que la computadora detecte una condición de interrupción, esta guarda lo que esta haciendo y transfiere el control al manejador de interrupciones que posteriormente regresa el control al punto donde se interrumpio.

Los microprocesadores de la familia Intel X86 manejan en forma más eficiente las interrupciones usando lectores de interrupción que les da ventaja sobre algunos procesadores que tienen restricción en sus manejadores de interrupciones.

Las interrupciones en las computadoras personales son relativamente fáciles para su manejo ya que estan diseñadas para un sólo usuario y sistemas de un solo proceso y debido a que la estructura de la interrupción es mucho más sofisticada.

Cuando una interrupción ocurre, el procesador se puede encontrar en cualquier estado. Un procesador esta diseñado para que éste termine siempre cualquier paso pendiente antes de responder a la interrupción. Cuando el procesador detecta una interrupción, este guarda el registro de banderas (la palabra del estado del programa), el apuntador de instrucciones (IP), y el registro de segmento de código (CS) en la pila y deshabilitando las interrupciones.

Cuando se ha guardado la información crítica, el procesador obtiene un número de 8 bits, el vector de interrupciones. Los vectores de interrupción son apuntadores de los programas que manejan funciones especificas. Este número que se obtiene del dispositivo de interrupción, indica la dirección en la tabla de vectores del programa que procesa la interrupción.

La familia de procesadores INTEL X86 define en sus primeros 25 bytes de memoria a la tabla de vectores de interrupción. El programa que procesa la interrupción se llama manejador de interrupciones.

El procesador multiplica el vector de interrupciones por 4 para obtener un desplazamiento de dirección, busca en el segmento 0000h para encontrar el vector. Esta dirección del vector se carga en el registro CS = IP, y el procesador continúa su operación desde la nueva localidad apuntada por el registro CS = IP.

Después de que el procesador esta bajo el control del manejador de interrupciones, éste controla al procesador. La mayoría de los manejadores primero rehabilitan la interrupción para que las interrupciones de alta prioridad puedan ser atendidas. También cuentan con registros para ejecutar sus propias operaciones lo mas rápido posible. Para algunos dispositivos se debe pasar una señal de reconocimiento para que el dispositivo se de cuenta de que es atendido.

Los manejadores de interrupciones deben ser diseñados para que sean lo mas rápido posible. La mayoría de ellos se codifican en lenguaje ensamblador para asegurar que la rutina sea lo mas rápido posible. En sistemas de computo mas grandes y potentes, los manejadores generalmente se codifican en lenguajes de alto nivel como "C" para hacerlos lo más simple posible.

En las computadoras personales se pueden codificar manejadores en "C" pero, las interrupciones de tiempo crítico deben ser manejadas en la forma más eficiente, es decir se deben escribir en lenguaje de bajo nivel.

Las interrupciones que se "disparan" con el controlador programable de interrupciones 8259A (PIC) mandan una señal de fin de proceso al PIC cuando se completa el proceso. Todas las interrupciones deben restaurar el estado de la máquina, refrescando los registros previamente salvados, y luego ejecutando la instrucción de regreso (IRET) para restaurar el registro de banderas, el registro IP y CS a los valores previos a que la interrupción ocurriera.

Vectores de interrupción.-

La tabla de vectores de interrupción se almacena en la parte baja de la memoria de sistema en los últimos 1024 bytes. Esta tabla utiliza 4 bytes por interrupción, haciendo un total de 256 vectores de interrupción. Los 4 bytes se componen del número de segmentos y el desplazamiento del manejador de interrupciones para una función determinada o para la dirección de una tabla de valores, como puede ser la tabla de caracteres gráficos que es apuntada por la función 1Fh.

Cuando se manejan vectores de interrupción se deben tomar todas las precauciones debido a que si estos son afectados en forma diferente a la deseada puede causar efectos irreversibles.

El DOS permite manejar de una forma segura el cambio de los vectores de interrupción usando la interrupción INT_21h función 25h (establecer el vector de interrupción) y 35h (obtener el vector de interrupción).

Para establecer un vector de interrupción, se deben seguir los siguientes pasos:

1. Usar la función 35h para obtener el valor de vector actual y almacenarlo para uso posterior para restaurar la interrupción y cambiarlo a otra rutina de interrupción.
2. Usar la función 25h para establecer el nuevo vector.

Se recomienda el uso del manejador de interrupciones bajo ciertas circunstancias que involucren alguno de los siguientes casos :

- Cuando se tiene que tomar el control de una interrupción para evitar que se de un mal funcionamiento del sistema.

- Por ejemplo, cuando el sistema detecte una división por cero se debe evitar que el procesador pierda el curso y se pierda su control.

- Cuando se quiera ligar una cadena de interrupciones. Por ejemplo, un programa residente que se ejecute basado en algunas teclas predefinidas.

- Cuando se requiera controlar el puerto serie. Como se mencionó antes, el DOS no proporciona la forma de manejo adecuada , en este caso para los puertos serie.

Se recomienda que cuando el caso lo amerite, se debe codificar la interrupción en un lenguaje de alto nivel. Cuando se necesite en la aplicación un manejador de alta velocidad, esta se debe codificar en un lenguaje de bajo nivel. El depurar un programa en lenguaje ensamblador resulta más difícil que en un lenguaje de alto nivel, pero las ventajas son indiscutibles. En estos casos, el programador debe tener conocimientos profundos sobre el funcionamiento del hardware de la máquina.

Cuando se codificaba un manejador de interrupciones, se debe tener cuidado al usar las funciones del DOS . El DOS no es reentrante, y si se interrumpe llamando alguna función del DOS nuevamente, se puede provocar un mal funcionamiento del sistema.

Una forma de llamar a las funciones del DOS es hacer por ejemplo, que el manejador de interrupciones realice la copia de los datos al buffer de memoria. Se puede activar una bandera que se puede reconocer por el programa que esta accesando el sistema en ese instante para realizar algún procesamiento adicional, el cual puede involucrar llamadas del DOS.

Existe una función que no está documentada para INT 21h (la función 34h) que regresa un apuntador en el registro ES = BX. Este apunta a la bandera de ocupado del DOS, llamada bandera InDOS. Cuando comienza la bandera es cero, ninguna función del DOS se ejecuta. De acuerdo con (DOS programmer's reference), no existe documentación de la función 34h. Se cree que ésta fué encontrada por medio de los que llaman "Hackers" (algo así como descifradores de información).

Los programas residentes checan la bandera cuando la tecla latente se presiona. Si la bandera no es cero, el programa residente activa la bandera de la llave latente en el programa residente. Los programas residentes que hacen esto, checan el estado de InDOS 18.2 veces por segundo hasta que la bandera se limpia. Cuando InDOS se limpia y la bandera de llave latente se activa, el programa residente comienza sus operaciones.

Cuando se trabaja con interrupciones, se debe asegurar de que se siga esta regla muy simple: siempre asumir que los otros programas estarán involucrados. Por ejemplo, nunca se debe activar el vector de interrupciones en forma directa. La INT 21h, función 25h se proporciona para este propósito y prevenir las mezclas entre programas al activar los programas.

Cuando el programa termina, este debe limpiar todas las interrupciones que ha activado.

Problemas con programas residentes .-

Los programas residentes pueden interrumpir al procesador en cualquier momento. El conocer lo que la máquina esta realizando cuando se toma el control por parte del programa residente es imposible.

Debido al diseño que desde el principio se dio a las computadoras personales, el manejo de varias tareas se complica. El BIOS (Basic Input-Output System) y el DOS (Disk Operating System) almacenan una gran cantidad de informacion en tablas globales. Los resultados intermedios de las entradas de datos y de las operaciones usan áreas temporales de memoria compartidas.

El problema que se presenta cuando se interrumpe al sistema por parte de un programa residente, es que el control puede estar en ese momento en poder del DOS. Si después de la interrupción se llama nuevamente al DOS, posiblemente causará un mal funcionamiento interno en el sistema que será imposible controlar.

Cuando se realizaron los primeros programas residentes estos causaron muchos problemas al sistema operativo (DOS) y a otros programas residentes. Todos estos problemas sucedian debido al poco conocimiento y experiencia que se tenía sobre programas residentes.

Las reglas para escribir programas residentes se elaboraron en base a la experiencia y no siguien algún fundamento teórico. Estas reglas son realmente recomendaciones de aquellos programadores que han descubierto ciertos comportamientos internos al sistema de los cuales no se tenía conocimiento, ni mucho menos documentación alguna.

Algunas de estas reglas se enumeran a continuación :

- Nunca llamar funciones del DOS a menos que no se tenga otra opción para realizar lo que se quiere. El tener que accesar archivos del sistema es la razón principal para recurrir al DOS.

Si se requiere realizar un acceso de entrada/salida se recomienda lo siguiente :

1. Usar otra forma alternativa para las funciones de consola de entrada/salida (INT 21H, funciones 01h-0ch).

2. Monitorear la bandera InDos. Cuando esta bandera no sea cero, el DOS se encuentra ejecutando una interrupción del tipo INT 21H. No se debe de correr el programa cuando esta bandera no sea cero.

3. Monitorear la INT 28H. Esta interrupción indica que aunque el DOS este realizando funciones INT 21H, esta se encuentra en un estado de "ocupado" para accesos de entrad/salida de la consola.

- Proporcionar una función de chequeo que permita al programa residente indicar si se encuentra instalado.

  El ligar un vector de interrupción que no sea usado al programa residente nos permite que se cheque la presencia de una copia previa del código ejecutable en memoria.

- Dar una firma dentro del código ejecutable para indicar que el programa residente se encuentra presente.

- Siempre se debe suponer que se encuentran presentes otros programas residentes. Relacionar cualquier interrupción que se use por el programa residente, pasandole el control al vector de interrupciones que el programa residente encontró cuando inició.

- Usar una pila propia en lugar de una controlada por el programa de interrupciones.

ORGANIZACION DE DISCOS FLEXIBLES

Anatomía de discos flexibles.-

Dentro del diseño del protector a través del uso de hardware, se encuentra una parte relacionada con la detección de accesos a disco flexible.

Para la correcta manipulación del manejador de discos, es necesario conocer la anatomía y funcionamiento del mismo. Esto tomando en cuenta que, un adecuado conocimiento del manejador nos permitirá programar, a través del uso de interrupciones, un manejador alterno "modificado".

Los medios más comunes de almacenamiento secundario que se utilizan actualmente son los discos flexibles y los discos duros. Los discos duros y flexibles difieren en capacidad para almacenar datos pero se manejan de la misma forma.

Es importante conocer la forma como se organiza la información en un disco de almacenamiento secundario ya que la correcta manipulación que se tenga sobre el disco es importante para el diseño del protector de software.

Se puede decir que un disco almacena un conjunto de archivos que pueden ser accesados a través de un directorio. Estos archivos existen sólo como una interpretación del sistema operativo .

Cuando se formatea un disco, el sistema operativo proporciona la estructura de archivos familiares al DOS. El sistema operativo proporciona un índice o directorio de archivos, así como una tabla para determinar donde se localizan estos (la FAT : de File Allocation Table). El sistema operativo almacena información especifica del disco, en el registro de Boot, el cual incluye un programa de inicio aún en discos que no contienen el sistema operativo.

Los discos flexibles cuentan generalmente con dos lados donde se puede almacenar o leer información por medio de una cabeza de lectura/escritura. Los discos duros cuentan generalmente con 2 o 4 discos o platos con superficies de almacenamiento dobles.

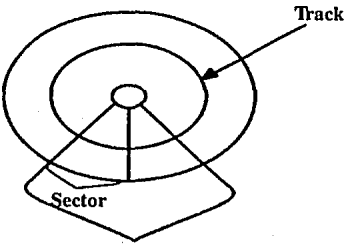


Fig. 4.1. Superficie de un disco flexible

En un manejador de disco la cabeza de lectura/escritura se lee usando un motor de pasos que marca por cada paso una pista en el disco. En un disco duro se tienen multiples cabezas que accesan la información en los discos o platos multiples accesando un cilindro (o grupo de pistas) por paso.

El programa que formatea divide las pistas en sectores de 512 bytes para crear segmentos de disco que se puedan manipular más fácil, de 8 a 9 sectores por pista en un disco flexible; 17 sectores o más por pista en un disco duro.

El sistema operativo asigna espacio a un archivo en unidades llamadas "clusters". Un "cluster" puede tener de 2 a 8 sectores, dependiendo del tipo de disco. Cuando algún archivo necesita más espacio, el sistema operativo asigna los "clusters" necesarios (Ver figura 4.1).

Un disco se divide en cinco áreas importantes :

- La tabla de particiones
- El registro Boot (o de arranque)

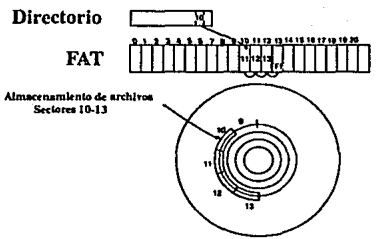


Fig. 4.2 Encadenamiento de clusters en la

- La tabla de alojamiento de archivos (FAT)
- El directorio
- El espacio para datos

La tabla de particiones .-

Todo disco duro contiene un registro de arranque maestro (Boot) que reside en el cilindro (pista) 0, cabeza (lado) 0, sector 1. este registro de arranque es el responsable de leer y descifrar la tabla de particiones del disco contenida al final del registro maestro de arranque (Boot).

El registro de arranque (Boot).-

Cuando el sistema ha determinado donde localizar el registro de arranque para la partición correspondiente en el disco, el BIOS carga el registro de arranque en memoria.

El registro de arranque bifurca a la dirección de arranque de operación del sistema y posteriormente carga el sistema operativo.

Tabla de alojamiento de archivos (FAT).-

La tabla de alojamiento de archivos (FAT) es una área usada por el sistema operativo para manejar el acceso al área de datos del disco. El sistema operativo utiliza la FAT para indicar que parte del disco es utilizada por un archivo.

La FAT esta compuesta por una serie de "bytes" usados para registrar el estatus de cada "cluster" en el manejador de disco. Existen códigos para indicar si un "cluster" esta en uso, disponible o dañado (ver fig. 4.2).

DISEÑO DEL HARDWARE

La segunda parte que involucra el diseño del protector se basa en el uso de un secuenciador AMD2910 del tipo Bit-Slice. Este dispositivo permite controlar las operaciones que se realicen en el circuito en forma externa, además de mantener comunicación con la computadora personal a través del puerto paralelo.

La importancia que juega éste circuito en el diseño es primordial, ya que las instrucciones de microprograma fueron diseñadas para cumplir con los fines particulares del proyecto.

Debido a que la aplicación que se le va a dar esta relacionada con la seguridad de la información, es importante que el microprocesador tenga una orientación muy específica y particular. Además, los circuitos "Bit-Slice" son una opción adecuada ya que dan gran libertad al diseñador para que éste utilice su creatividad.

PROCESADORES BIT-SLICE

Una evolución de los microprocesadores han sido las procesadoras "BIT-SLICE". Para algunas aplicaciones de propósito general, las unidades de procesamiento central tales como el 8088 y el 6800 no son tan rápidas ó su conjunto de instrucciones de máquina no es el adecuado para dicha aplicación. Para aplicaciones muy específicas, algunos fabricantes producen dispositivos con los cuales es posible instrumentar una unidad de procesamiento central. Un ejemplo de tales circuitos lo forman la familia AMD2900 de Advanced Micro Devices.

Esta familia incluye unidades aritméticas de 4 bits,multiplexores, secuenciadores, y otros dispositivos útiles para el diseño de microprocesadores.

El término "SLICE" del inglés significa rebanada y se le dá este nombre porque con varias "rebanadas" conectadas en paralelo se pueden diseñar microprocesadores con palabras de 8 bits, 16 bits o 32 bits. Esto es, un diseñador puede añadir tantas rebanadas como lo requiera para su aplicación.

Este tipo de dispositivos eletrónicos no sólo permite al diseñador elaborar su propio hardware, sino que además permite la instrumentación del conjunto de instrucciones o microcódigo.

Los BIT-SLICE permiten la elaboración de instrucciones complejas o simples dependiendo lo que se requiera. Cabe señalar que el grado de complejidad depende de número de ciclos de tiempo que tenga cada instrucción de máquina. De lo anterior, se puede afirmar que mediante el uso de estos microcontroladores se pueden diseñar arquitecturas de microprocesadores del tipo RISC o CISC.

La función principal del microprogramado es la de dar un medio simple, flexible y relativamente barato de control en una computadora. La ventaja que proporciona el control microprogramado es que permite instrumentar gran variedad de instrucciones, con un conjunto de instrucciones particular que aplica sólo para una determinada arquitectura.

El diseño del protector de software basa su organización en un esquema horizontal. Una organización horizontal es aquella que basa su diseño de codificación en varias líneas de control para diferentes recursos; todo esto incluído en una simple microinstrucción.

Al esquema que basa su organización en pocas líneas de control se le llama de organización vertical.

Un esquema con organización horizontal se recomienda cuando la velocidad operativa de una computadora es un factor crítico y donde la estructura de la máquina permite el uso paralelo de un gran número de recursos.

El esquema vertical tiene unas velocidades operativas menores, debido a que se necesitan más microinstrucciones para ejecutar las funciones de control. Sin embargo, se necesitan menos bits para cada microinstrucción. Esto no quiere decir que el número total de bits almacenados en la memoria de microprograma sea menor. El factor significante en este caso es la reducción de hardware en paralelo para el manejo de la ejecución de instrucciones de microprograma.

Actualmente el término "BIT-SLICE" se usa para describir a los dispositivos que realizan operaciones de procesamiento que no son capaces de funcionar en forma independiente y que requieren de control externo. Algunos ejemplos son los secuenciadores, las unidades aritméticas lógicas,procesadores de punto flotante, controladores de interrupciones, multiplicadores, controladores de acceso directo a memoria, etc.

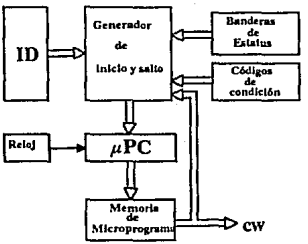
Los dispositivos "BITE-SLICE" son usados principalmente como controladores ó bien como microprocesadores. Debido a su velocidad y flexibilidad los "BITE-SLICE" son mas usados en aplicaciones de alta velocidad como puede ser el procesamiento de señales.

Cuando se comparan los micropocesadores normales de aplicación general con los "BIT-SLICE" para un diseño específico, su comparación se dificulta. Sin embargo, en cuestiones de velocidad cuando se ejecutan operaciones simple, las ventajas de los "BIT-SLICE" son superiores.

Un microprocesador 68010 de 12.5 MH le lleva 400 ns sumar dos registros. Un diseño usando un circuito de la familia AMD 29300 con un reloj de 8 MH lo podría hacer en 25 ns. Cuando se considera la funcionalidad con punto flotante, la diferencia en funcionalidad puede ser aún más impresionante.

Las ventajas que se obtiene al usar el circuito AMD2910 en el diseño del protector son obvias ya que la velocidad es importante cuando se tiene un sistema que debe actuar transparente al usuario.

Fig. 4.3. Unidad de control de microinstrucciones



CONTROL MICROPROGRAMADO

La microprogramación es muy parecida a la programación de software ya que se usan estructuras de hardware semejantes a la secuenciación de un programa ordinario. ahora bien, si se compara con un lenguaje de bajo nivel, se puede observar que las diferencias son pocas.

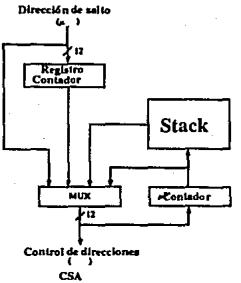


Fig. 4.4. Organización básica del secuenciador

Las microinstrucciones se almacenan en forma secuencial en la memoria de microprograma ó almacenamiento de control. Se usa un contador de microprograma (μPC) para apuntar a las direcciones de memoria. Este contador se incrementa de 1 en 1 para traer la siguiente instrucción o accesar datos almacenados en memoria.

Debido a que cada instrucción de máquina sigue una microrutina, se debe de dar una dirección inicial que apunte a esta microrutina en un registro de instrucciones (IR).

La organización de la unidad de control que se muestra en la figura 4.3; contiene un contador de microprograma que direcciona a la memoria de microprograma la cual tiene como salidas la dirección de la siguiente microinstrucción de la microrutina y una palabra de control. Se muestra también, el registro de instrucciones (IR) el cual tiene la dirección de una microrutina, la cual se mapea por medio del generador de instrucciones.

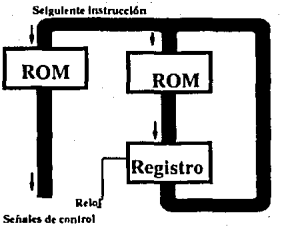


Fig. 4.5. Una memoria de sólo lectura (ROM) y un registro, forman la máquina de microcódigo más simple.

Este generador de instrucciones puede dar saltos condicionales o incondicionales por medio del registro de banderas. La figura 4.3 muestra a detalle estas características.

El diseño básico del protector de harware se basa en el secuenciador con control microprogramado AMD2910. A la salida del secuenciador se genera una dirección de control de almacenamiento de una de 4 fuentes posibles (ver figura 4.4). Se usa un contador microprogramado para direccionar secuencialmente. En cualquier tiempo, la salida de este contador puede conectarse a una pila para permitir llamadas a subrutina dentro del microprograma. Los saltos se instrumentan dando una dirección de salto a la entrada del secuenciador. Además, el AMD2910 tiene un registro con iteración que se usa como contador que almacena direcciones temporales y para simplificar la instrumentación de los ciclos repetitivos del microprograma.

Los esquemas de organización horizontal y vertical representan los dos extremos de organización para el control microprogramado. Existen otros tipos de organización en una computadora que usan esquemas intermedios y que se pueden describir como horizontales o verticales debido a su organización mezclada.

Actualmente, los procesadores microprogramados estan teniendo gran impacto en muchas aplicaciones para distintos propósitos, principalmente como resultado de los avances en la tecnología VLSI (de las siglas en inglés para los circuitos electrónicos de escala de muy alta integración). El desarrollo de los circuitos "BIT-SLICE" es un proceso significante en esta área.

Las máquinas de estados, que son los procesadores más simples y por tanto más rápidos, se pueden construir con lógica de "hardware" y algunos registros. Cuando los estados son pocos el diseño es muy simple así como su construcción, pero cuando los estados son muchos, el diseño y la construcción se complica.

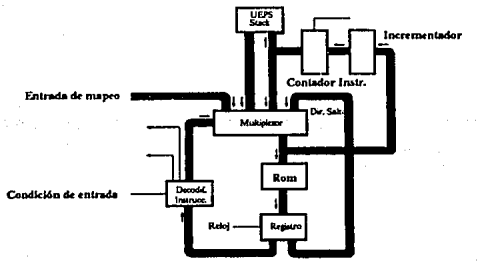


Fig. 4.6. La gráfica muestra un secuenciador de proposito general en donde se muestra que un campo de la Rom determina la siguiente instrucción a ejecutar.

El uso de microcodificación simplifica el diseño, permitiendo que las máquinas de estados con varios miles de estados posibles se produzcan con relativa facilidad. La configuración más simple para construir una máquina de microcódigo consta de una memoria de solo lectura (ROM) y un registro.

Las salidas de la ROM se dividen en dos partes, una para controlar algunos dispositivos externos y la otra para la dirección del siguiente estado en la ROM. Pero el uso de esta configuración es muy limitado debido a que la siguiente dirección no puede ser determinada en forma externa. (ver figura 4.5)

La figura 4.6 muestra un secuenciador de propósito general. Esté se compone de un multiplexor que mediante una instrucción de microcódigo proveniente de la ROM selecciona la fuente de donde obtendrá la dirección siguiente a ejecutarse en la ROM.

Las fuentes posibles son un contador de instrucciones el cual lleva una secuencia de las localidades en forma ascendente, el campo de la dirección siguiente de la ROM que se usa para saltar a otra secuencia, una entrada de mapeo usada para saltar a otra dirección nueva que se proporciona externamente, y una pila, donde un número determinado de direcciones, puede ser almacenado en forma UEPS (Ultimo que Entra-Primero que Sale) para el manejo de subrutinas.

Frecuentemente se debe tener un contador para poder realizar ciclos iterativos (IF,FOR,WHILE,GOSUB,GOTO Y RETURN) y de esta forma hacer el microcódigo más parecido a la programación de alto nivel y hacer que esto sea más sencillo y no tan difícil como parece ser a simple vista.

De esta forma, el controlador primitivo consiste de un secuenciador con una línea de entrada con condición simple, algunas ROMs conectadas en paralelo para tener un tamaño de microcódigo tan largo como se requiera, y una entrada de mapeo con las líneas de direccionamiento necesarias para controlar las ROMs.

Las ROMs se usan normalmente en diseños con microcódigo debido a que reducen el número de circuitos integrados necesarios. Las ROMs permiten que la codificación sea sencilla, además de producir señales de control "limpias" a lo largo de todo un ciclo de reloj.

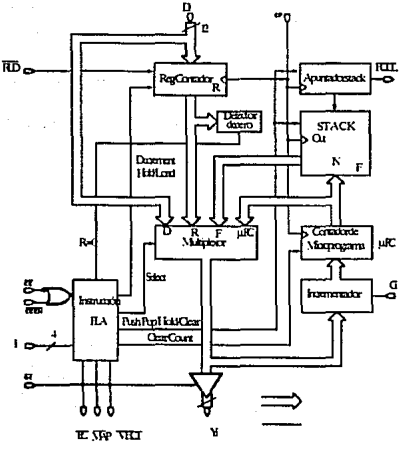


Fig 4.7. Diagrama de bloques interno del secuenciador

La mayoría de los dispositivos controlados por "bit-slice" tienen una arquitectura en donde las ROMs de microcódigo son las fuentes de sincronización en el diseño, minimizando los sesgos de reloj y simplificando los cálculos de tiempo.

No es ningún problema el tener diferentes partes de un circuito con diferentes cauces. Pero si el circuito que se diseña debe responder instantáneamente a algún estímulo externo, por ejemplo una interrupción, es importante que los elementos controlados con diferentes cauces, no tengan ninguna interación.

PROCESADOR AMD2910

El secuenciador microprogramado AMD2910 es un secuenciador de direcciones diseñado para controlar la ejecución de las microinstrucciones que se encuentran almacenadas en la memoria de microprograma (ver fig. 4.7).

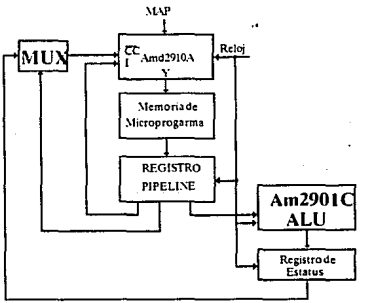


Fig. 4.8. Arquitectura de un nivel de pipeline usando el circuito Amd2910A

Este proporciona una gran variedad de instrucciones, condicionales casi todas ellas, dentro de un rango de 4096 instrucciones posibles que se pueden direccionar.

Tiene una pila del tipo UEPS (Ultimo que Entra, Primero que Sale) el cual permite realizar ciclos iterativos y llamadas a subrutinas; los niveles de llamados a subrutinas (rutinas anidadas) que permite el secuenciador es de nueve.

Durante cada microinstrucción, el controlador microprogramado proporciona una dirección de secuencia de cuatro fuentes posibles :

1.- El registro de microprograma (μPC).
Usualmente contiene la dirección de la instrucción siguiente.

2.- Una entrada externa directa (D).
Por la cual se pasan datos o direcciones.

3.- Un registro/contador (R).
Este permite retener información cargada en una microinstrucción previa.

4.- Una pila de nueve niveles (F).

El Amd2910A es un remplazo de alta velocidad del Amd2910. Este puede mejorar la velocidad del Amd2910 hasta en un 35%, además de tener una pila de más niveles.

Algunas características distintivas de éste secuenciador son:

- Tamaño de 12 Bits
  Direccionamiento hasta de 4096 palabras de microcódigo en un solo circuito. Todos los elementos internos tienen un tamaño de 12 bits.

- Contador iterativo interno.

  Contador tipo "preset" para repetición de instrucciones y contador de iteraciones.

- Cuatro fuentes de posible direccionamiento.

  las direcciones de microprograma pueden ser seleccionadas del contador de microprograma, del grupo de direcciones de salto, de la pila de entrada/salida de 9 niveles, o del registro interno.

- Dieciseis microintrucciones poderozas.

  Este microcontrolador puede ejecutar 16 instrucciones de control de secuencia, la mayoría de las cuales son condicionales y son entradas externas, estados de algún contador de ciclos iterativos, o ambas.

- Controles de habilitación de salida para tres fuentes de direccionamiento para saltos.

  Decodificador de construcción interna para habilitar dispositivos externos en el canal de direccionamiento de salto.

- Rápidez.

  El secuenciador Amd2910A proporciona ciclos de reloj de 100ns y es de 25 a 30% más veloz que el AMD2910.

La arquitectura recomendada por el fabricante se muestra en la fig. 4.8.

DIAGRAMA DE BLOQUES.-

Anatomía del protector .-

El circuito "Bit-Slice" es unidad central de proceso de protector. El bus de datos que se observa en el diagrama proporciona datos de la ROM de instrucciones o de los datos provenientes de la PC. Estas instrucciones pueden ser proporcionadas por las direcciones de las instrucciones que se obtienen del registro MAR (de Memory Adress Record) o de la PC de forma directa. A continuación se explica en forma detallada cada bloque que forma el protector.

Registro de instrucciones .-

Este registro almacena las direcciones del inicio de cada instrucción de microcódigo. Las instrucciones se cargan al registro de instruciones habilitando la linea LDRI; las salidas de este registro entran al registro de mapeo cuando se habilita la linea LDMAP. La información es tomada de la memoria de microprograma, la cual tiene el programa de control.

El registro de instrucciones puede alimentar datos a los registros internos del ALU AMD2901, proporcionar direcciones de salto o instrucciones al AMD2910 a través del registro de mapeo.

Registro de mapeo.-

El registro de mapeo recibe una señal de habilitación cuando el secuenciador AMD2910 requiere datos o direcciones de entrada. Tambien, éste cuenta con señales de habilitación OEMAP y de carga de datos LDMAP. La carga de datos se controla a través de un estado en la memoria de microprograma (LMAP) y se habilitan con una señal que viene directa del AMD2910 (MAP). Esta señal MAP es proporcionada por el AMD2910 cuando se ejecuta su instrucción interna JMAP.

Registro de datos.-

El registro de datos siempre lee los datos que vienen directamente de la PC, a través del puerto paralelo. Su salida esta normalmente habilitada y proporciona datos al ALU AMD2901. Los datos pueden ser instrucciones para el programa en microcódigo o bien pueden ser datos a ser encriptados. La ALU AMD2901 es la que se encarga de pasar la instrucción o el dato de acuerdo a la instrucción de microcódigo.

Secuenciador AMD2910.-

El secuenciador AMD2910 tiene tres señales. La señal MAP habilita el registro de mapeo para recibir datos de instrucciones o direcciones del registro de mapeo.

La señal de entrada CC es una señal binaria que es alimentada por un multiplexor de acuerdo a la selección de una variable a sensar. Es a través de esta señal que el circuito AMD2910 se acopla con la computadora personal.

La selección de la variable a través del MUX se hace por medio de señales de estados que se encuentran en la memoria de microprograma.

La señal de salida PL, habilita las memorias de microprograma y a los registros de "pipeline". La mayoria de las instrucciones de éste secuenciador habilitan esta señal debido a que el secuenciador casi siempre proporciona señales de control.

El AMD2910 recibe datos o direcciones del registro de mapeo o del registro de segmento ; éste registro proporciona la dirección de salto para las instrucciones de microprograma.

El AMD2910 tiene como salidas, lineas que direccionan a las memorias de microprograma. Estas direcciones permiten obtener los estados de acuerdo a la carta ASM (de Algoritmic State Machine).

El Secuenciador puede seleccionar 16 posibles instrucciones para microprograma de los estados que manejan la instrucción siguiente que se alimenta al AMD2910 (I0-I3). Estas instrucciones pueden habilitar diferentes señales de control para dispositivos externos de entrada o salida según el diseño.

Memorias de microprograma.-

En estas memorias de sólo lectura (ROMs) se almacenan todos los estados que se definieron en el diseño. Estas memorias se conectan a los registros de "pipeline", los cuales mantienen las salidas de las ROMs en todo el ciclo de reloj del sistema. Se cuentan con cuatro memorias de microprograma, la primera, contiene la siguiente instrucción del secuenciador y las señales del multiplexor para seleccionar la fuente para la señal CC del AMD2910, la segunda, contiene la instrucción del ALU AMD2901, la tercera, contiene las señales de control de los registros de trabajo de circuito encriptador, y la cuarta, contiene instrucciones o direcciones de saltos incondicionales.

Registros "Pipeline".-

Los registros "pipeline" mantienen el estado de las salidas de las memorias de microprograma durante todo el ciclo de reloj.

Los registros se habilitan con la señal PL que viene del secuenciador AMD2910 y sus salidas son los estados del la máquina algorítmica que van a diferentes dispositivos o registros de trabajo.

Registro de segmento.-

El registro de segmento proporciona direcciones de salto para el manejo interno de subrutinas. Este permite direccionar hasta 8 lineas. Los 8 estados provienen de los registros de "pipeline".

La señal de carga siempre esta habilitada y la de salida OERS controla por microprograma. Las salidas deben estar en alta impedancia cuando la señal MAP del secuenciador se encuentra habilitando las salidas del registro de instrucciones.

Unidad Aritmética Lógica AMD2901 .-

Se tienen dos circuitos AMD2901 conectados en paralelo para manejar una palabra de 8 Bits. Reciben datos de entrada de el registro de datos o del registro de instrucciones. El registro de instrucciones proporciona datos para cualquiera de los registros internos A ó B del AMD2901.

Las instrucciones, los registros fuentes y los registros destinos del ALU se manejan a través de 9 lineas de estado (ALU 0-8) que vienen de los registros de "pipeline".

Las 8 salidas de las unidades aritmética lógicas (OEALU) alimentan al registro MAR proporcionandole datos de encripción o instrucciones provenientes de la PC para el programa de microprograma.

La ALU tiene conectados los bits de salida más significativos de un registro interno de trabajo a los menos significativos, esto, para poder efectuar operaciones de desplazamiento.

Registro de direcciones de memoria MAR.-

El registro MAR (de Memory Address Register) toma las direcciones de la siguiente instrucción de microprograma del ALU AMD2901 y se la proporciona a la memoria que contiene a las instrucciones de microprograma.

Tiene una señal de carga LDMAR y otra de habilitacion OEMAR. Estas señales son controladas por microprograma.

Funcionamiento básico.-

El circuito recibe datos de la PC los cuales pueden ser datos para el microprograma, direcciones de salto o instruciones de microprograma. Esta es sin duda una de las ventajas de este diseño para mantenerse confidencial.

En primer término, el circuito recibe una instrucción de la PC la cual le indica que tome el control, después, el circuito reliza las operaciones de encripción que correspondan en forma secuencial, hasta regresar el control a la PC. El programa en microcódigo interectua con la PC para tomar datos o posibles direcciones nuevas de microprograma.

En la Fig. 4.9. se muestra la forma en que se interrumpe un programa que se esta ejecutando cuando se encuentra la rutina protectora residente en memoria.

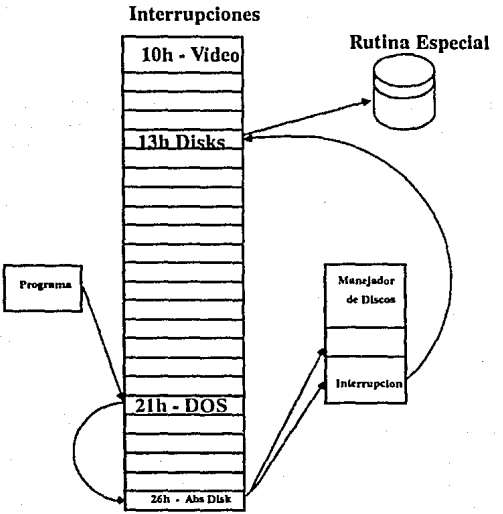


Fig. 4.9. Llamada desde un programa para acceso a disco.

El programa se interrumpe a un nivel del DOS y la rutina para accesar discos se llama del BIOS, para este caso INT 13h. Esta rutina se modifica (ver apendice) para que interactue con el circuito externo y encripte la información que se almacena en el "buffer" especial de la interrupción. Una vez que se ha realizado la encripción, el programa y la interrupción 13h continuan su operación normal.

A continuación se resumen los pasos generales que el sistema debe seguir.

1. El programa en software se carga en memoria y permanece residente.

2. La rutina especial, residente en memoria, detecta un acceso al disco.

3. El programa "desvia" la interrupción 13h a una rutina especial.

4. La rutina especial toma la información del "buffer" del manejador de discos del BIOS (INT 13h) y la encripta a través de la comunicación con el circuito protector externo.

5. El control se regresa a la rutina normal del manejador de discos (INT 13h) para que realice el acceso al disco en forma normal.

6. El control se regresa al programa o a la aplicación que llamo al la rutina del manejdor de discos del BIOS.

CONCLUSIONES

Durante el desarrollo del presente trabajo se mencionó la importancia de la protección de información de acuerdo a la sensibilidad de la misma; esta se refiere a la complejidad en un sistema de protección de datos. El grado de sofisticación de un sistema de seguridad no necesariamente se refiere a los aspectos técnicos que se utilizan para construir un protector determinado, se deben tomar en cuenta además, los procedimientos de seguridad que se establezcan para instrumentar un programa de seguridad de datos. Debido a esto, se puede afirmar que un programa de seguridad es prioritario al instrumentar cualquier sistema de seguridad.

Como se explicó durante el desarrollo de éste trabajo, se debe evaluar en forma detallada en que proporción superan nuestras ventajas a nuestras desventajas para que de esta forma se pueda escoger la mejor forma de proteger lo que se considera el activo más importante en una compañía, la información. Es importante recalcar que el papel que juegan los datos difieren en cada compañía dependiendo del rubro de la misma; esto es, el grado de automatización que tenga un negocio lo hará más dependiente de su información..

Para poder hacer una mejor evaluación del procedimiento de seguridad que se instrumenta es importante saber contra que o contra quienes se debe proteger la información, ya que ésto permitirá escoger un procedimiento adecuado que permita mantener la integridad de los datos y al mismo tiempo no obstruya las tareas diarias del usuario encargado de mantener la información.

La encripción de datos es una de las formas de proteger la información más eficientes y por eso es conveniente afirmar que un buen sistema de seguridad debe incluirla. Esto se debe a que los sistemas de encripción computarizados son virtualmente imposibles de "romper". La regulación de estos sistemas de encripción actualmente se encuentran en una controversia al permitir el uso indiscriminado de los sistemas públicos para encriptar. Como se mencionó en éste trabajo la eficiencia de un sistema de encripción no esta en la complejidad de su algoritmo, sino en la confidencialidad de la llave que se maneje. El hecho de permitir un uso indiscriminado de un sistema de encripción evita que las autoridades puedan consignar a presuntos criminales de la información, esto debido a que estos mantienen los datos plagiados encriptados, y por tanto, no se tendrían evidencias suficientes para poder consignarlos. Por otro lado la ventaja de no regular la encripción permitiría usar las encripción como una "comodidad" vital para la protección de la información. Además, la encripción es una forma natural de protegernos. Sin embargo, la encripción actualmente es considerada una parte esencial en un sistema de protección y simplemente debe ser complementada con algunas otras técnicas de seguridad.

Los controles de acceso juegan un papel importante en la actualidad. Las palabras clave de acceso son utilizadas principalmente por sistemas grandes. Es aqui en donde se puede pensar en complementar algunas técnicas de encripción con las técnicas de control de accesos; de esta forma se tendrian claves de acceso que pueden ser utilizadas como llaves de encripción. Existen miles de formas en que se podrian combinar las diferentes formas de protección de datos con alguna forma de encripción y es conveniente evaluar cual es la forma que más se adapta a las necesidades del sistema.

Se deben tomar en cuenta las estadísticas que se mostraron en éste trabajo para concluir que los problemas de seguridad de la información deben considerarse internos a la organización. Si se considera que sólo el 5 % de los ataques en contra de la información provienen del exterior y de forma premeditada; considerar una alta inversión en seguridad contra ladrones especializados, externos a la organización, sería injustificada.

Además, no se debe ignorar que la protección debe estar encaminada a la seguridad de la información y no al equipo o "hardware". Esto es, el activo que más importa en éste caso son los datos, que en algunos casos se pueden considerar irrecuperables si se dañan o extravian. Por tanto, la pérdida de un equipo no es equiparable a la pérdida de información, obviamente por el valor de recuperación que cada caso significa.

Los problemas más comunes de seguridad y sus formas de atacarlos se pueden resumir en los siguientes :

. Accesos no autorizados.- Este es uno de los problemas más comunes se combaten a través del uso de claves de acceso y mediante un rastreo permanente.

. Mecanismos ineficaces en el control de accesos.- Se deben evaluar procedimientos y utilerías que permitan controlar este tipo de accesos, que además, faciliten la labor cotidiana del usuario.

. Virus .- Se deben instrumentar procedimientos eficientes y que concienticen al usuario para prevenir y erradicar los virus informáticos. Los procedimieintos contra virus informáticos deben existir sin ser estos considerados como la solución a éste problema.

. Instalación no autorizada de equipo y modems.- Se debe tener control total sobre el equipo que se tiene, así como realizar auditorías periódicas de forma interna y externa en forma aleatoría. Se debe contar además con un equipo de seguridad física para el equipo.

. Software no autorizado.- Se deben instrumentar políticas de seguridad organizacionales que impidan el uso de software no autorizado y que obliguen a tener auditorías permanentes.

. Políticas y procedimientos de seguridad anticuados .- Se deben generar planes de actualización sobre políticas de seguridad con el fin de que posibles procedimientos obsoletos impidan el libre trabajo productivo de un usuario. Para esto es necesario hacer estudios que evaluen las ventajas y desventajas de tales procedimientos.

. Expansión desconocida e incontrolada de la red.- En éste caso, se den establecer planes de crecimiento para redes de área local, así como los procedimientos que deberan aplicarse al momento de expanderse dicha red.

. Información no etiquetada con advertencias de control .- Para controlar éste problema de seguridad es necesario instrumentar técnicas y procedimientos para organizar y documentar información.

Es importante mencionar que la consistencia y el seguimiento permitirán que un buen procedimiento de seguridad tenga éxito o no.

Las pautas que se deben tener para determinar los requisitos de seguridad que son recomendables son las siguientes :

- Establecimiento del valor de la información.
- Establecimiento de la propiedad de la información.
- Determinación del grado de confidencialidad o integridad de los datos.
- Análisis de amenazas y puntos vulnerables.
- Consecuencias de la pérdida de la información.
- Evaluación de ventajas y desventajas del rendimiento, seguridad y costos.
- Evaluación de riesgos.

Estas pautas ayudan a establecer el mejor método de seguridad mediante la evaluación de ventajas y riesgos. Es importante tener en cuenta que el mejor sistema de seguridad es el más adecuado a las necesidades de cada organización. Es decir, no se puede tener un sistema de seguridad que cubre las necesidades de una compañía de seguros, instalado en una universidad

El secuenciador AMD2910 que se utilizó en el diseño del protector tiene la ventaja de permitir que el diseñador cree su propio conjunto de instrucciones en microcódigo. Este conjunto de instrucciones puede ser creado bajo la filosofía CISC (Complex Instruction Set Computer), es decir, bajo un concepto en el cual el número de ciclos de reloj por instrucción es mayor debido a la complejidad de una instrucción en microcódigo. El diseñar bajo CISC proporciona las ventajas necesarias para crear un sistema seguro, ya que se tiene una mayor velocidad de procesamiento y las intrucciones pueden permanecer confidenciales. Obviamente una desventaja de diseñar bajo CISC es que la aplicación debe ser muy particular, pero si hablamos de seguridad, esta desventaja se vuelve ventaja.

Si a un diseño tipo CISC se le agrega un diseño con PLA (Programable Logic Array) en lugar de utilizar una EPROM la posibilidad de que este sistema sea violado se reduce debido a la confidencialidad que se tiene al diseñar con un PLA.

El circuito AMD2910 es altamente recomendado para diseñar sistemas en tiempo real. La recolección de datos mediante sensores es una parte importante en sistemas de éste tipo; es aquí en donde el uso de un secuenciador como el AMD2910 podría ser utilizado en lugar de algunos procesadores comerciales de proposito general. Las ventajas son mayores, ya que un sistema que opera en tiempo real los accesos a memoria son importantes en cuanto a velocidad se refiere; además, la codificación en microcódigo hace más eficientes las operaciones de entrada/salida tanto en los sensores como en la relación PC-recolector.

Es importante mencionar que bajo estas circunstancias el diseño del protector, explicado en este trabajo, tiene ventajas significativas en cuanto a seguridad se refiere, ya que gracias a la flexibilidad con la que puede ser configurado el circuito AMD2910 esto representa un obstaculo para que el sistema sea violado.

La información es parte vital de cualquier ser humano. La carrera armamentista, espacial y comercial, por no mencionar otras, se rigen por el poder que proporciona la información y su correcta administración. La pérdida y mal manejo de los datos almacenados en una computadora puede traer resultados desastrosos difíciles de imaginar.

La labor de toda organización es invertir lo necesario en la seguridad de la información sin escatimar recursos de ningún tipo si se tiene en juego lo más valioso que se posee, que sin duda alguna es la información.

I0-I3	MNEMONICO	NAME	REG/CONT RCONTENTS	FAIL		PASS		REG/CNTR	ENABLE
				CCEN =L AND $\overline{CC}$ = H		CCEN = H OR $\overline{CC}$ =L			
				Y	STACK	Y	STACK		
0	JZ	JUMP ZERO	X	0	CLEAR	0	CLEAR	HOLD	PL
1	CJS	COND JSB PL	X	PC	HOLD	D	PUSH	HOLD	PL
2	JMPA	JUMP MAP	X	D	HOLD	D	HOLD	HOLD	MAP
3	CJP	COND JUMP PL	X	PC	HOLD	D	HOLD	HOLD	PL
4	PUSH	PUSH/COND LD CNTR	X	PC	PUSH	PC	PUSH	NOTE 1	PL
5	JSRP	COND JSB R/PL	X	R	PUSH	D	PUSH	HOLD	PL
6	CJV	COND JUMP VECTOR	X	PC	HOLD	D	HOLD	HOLD	VECT
7	JRP	COND JUMP R/PL	X	R	HOLD	D	HOLD	HOLD	PL
8	RFCT	REPEAT LOOP CNTR <>0	<>0	F	HOLD	F	HOLD	DEC	PL
			=0	PC	POP	PC	POP	HOLD	PL
9	RPCT	REPEAT CL. CNTR <>0	<>0	D	HOLD	D	HOLD	DEC	PL
			=0	PC	HOLD	PC	HOLD	HOLD	PL
10	CRTN	COND RTN	X	PC	HOLD	F	POP	HOLD	PL
11	CJPP	COND JUMP PL & POP	X	PC	HOLD	D	POP	HOLD	PL
12	LDCT	LD CNTR & CONTINUE	X	PC	HOLD	PC	HOLD	LOAD	PL
13	LOOP	TEST END LOOP	X	F	HOLD	PC	POP	HOLD	PL
14	CONT	CONTINUE	X	PC	HOLD	PC	HOLD	HOLD	PL
15	TWB	THREE WAY BRANCH	<>0	F	HOLD	PC	POP	DEC	PL
			=0	D	POP	PC	POP	HOLD	PL

X = DON'T CARE H = HIGH L = LOW

CONJUNTO DE INSTRUCCIONES DEL SECUENCIADOR AMD2910

NOTE 1: SI $\overline{CCEN}$ = L Y $\overline{CC}$ = H, ENTONCES SE HACE UN HOLD DE LO CONTRARIO SE HACE UN LOAD.

APENDICE A

Am2910
MICROPROGRAM CONTROLLER

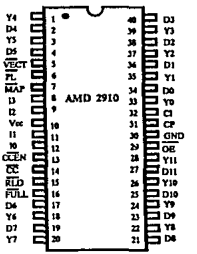
DESCRIPCION GENERAL

El controlador microprogramado Am2910 es un secuenciador de direcciones con el fin de controlar la secuencia de la ejecución de macroinstrucciones almacenadas en la memoria de microprograma. Además de la capacidad de acceso secuencialmente, esta permite saltos condicionales a cualquier microinstrucción dentro de su rango de 4096 micro-palabras. Una pila del tipo last-in, first-out permite el retorno de subrutinas y la capacidad de manejar ciclos iterativos; teniendo además nueve niveles de subrutinas.

Cuenta con un contador de ciclos de doce bits que le permiten direccionar las 4096 micro-palabras. Tiene cuatro fuentes posibles de direccionamiento que son el contador de microprograma, el bus de dirección de saltos, un stack de nueve niveles o del registro de trabajo interno.

Cuenta además con un poderoso set de instrucciones de 16 instrucciones de control de secuencia, la mayoría de las cuales reciben una condición de entrada externa, o del estado del contador de iteraciones o de ambos.

Este set de instrucciones se muestra en forma detallada en la parte de arriba.



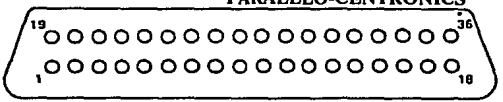
No

Existe

Pagina

APENDICE B

INTERFACE PUERTO PARALELO-CENTRONICS



INTERFACE IMPRESORA-PUERTO PARALELO

NO. DE PIN	SEÑAL	IN/OUT	EXPLICACION
1	DATA STROBE	IN	INDICA PRESENCIA DE DATOS. EL DATO SE LEE CUANDO EL FLANCO DE LA SEÑAL ES BAJO (LOW).
2-9	DATA 1-8	IN	DATOS EN 8 BITS. EL NIVEL ES ALTO PARA '1' Y BAJO PARA '0'.
10	ACKNLG	OUT	SEÑAL DE RECONOCIMIENTO DE LA IMPRESORA A LA SALIDA.
11	BUSY	OUT	SEÑAL QUE INDICA SI LA IMPRESORA PUEDE RECIBIR ALGUN DATO. SE PERMITE LA ENTRADA DE DATOS CUANDO LA SEÑAL ES BAJA.
12	PE	OUT	SEÑAL DE 'DC' QUE TIENE NIVEL ALTO CUANDO NO HAY PAPEL.
13,35	+ 5 V	OUT	ELEVAR HASTA + 5V POR RESISTENCIA DE 4.7 Ω
14	AUTO FEED	IN	SEÑAL QUE INDICA A LA IMPRESORA QUE CARGUE EL PAPEL
18	+ 5 V SOURCE	OUT	SALIDA DE 300 mA
31	INPUT PRIME	IN	CUANDO LA SEÑAL ES BALA LA IMPRESORA SE REPONE DE L ESTADO INICIAL
32	FAULT	OUT	SEÑAL DE 'DC' QUE ES BAJA CUANDO EL IMPRESOR SE ENCUENTRA FUERA DE LINEA
36	SLCT IN	IN	CUANDO ES BAJA EL IMPRESOR ESTA EN LINEA, Y CUANDO ES ALTO ESTA FUERA DE LINEA

NO

Existe

Pagina

APENDICE C

TABLA DE FUNCIONES DEL ALU AMD2901

FIG. 1 CONTROL DEL OPERADOR FUENTE ALU

MNEMONICO	MICROCODIGO				OPERADORES FUENTE ALU	
	I2	I1	I0	CODIGO OCTAL	R	S
AQ	L	L	L	0	A	Q
AB	L	L	H	1	A	B
ZQ	L	H	L	2	0	Q
ZB	L	H	H	3	0	B
ZA	H	L	L	4	0	A
DA	H	L	H	5	D	A
DQ	H	H	L	6	D	Q
DZ	H	H	H	7	D	0

FIG.2 CONTROL DE FUNCION DEL ALU

MNEMONICO	MICROCODIGO				FUNCION ALU	SIMBOLO
	I5	I4	I2	CODIGO OCTAL		
ADD	L	L	L	0	R PLUS S	R + S
SUBR	L	L	H	1	S MINUS R	S - R
SUBS	L	H	L	2	R MINUS S	R - S
OR	L	H	H	3	R OR S	R V S
AND	H	L	L	4	R AND S	R & S
NOTRS	H	L	H	5	R AND S	$\overline{R}$ & S
EXOR	H	H	L	6	R EX-OR S	R V S
EXNOR	H	H	H	7	R EX-NOR S	$\overline{R \vee S}$

X = DON'T CARE; B = REG. DIRECCIONADO POR LAS ENTRADAS B; UP = MSB, DOWN = LSB.

FIG. 3 CONTROL DE DESTINO DEL ALU

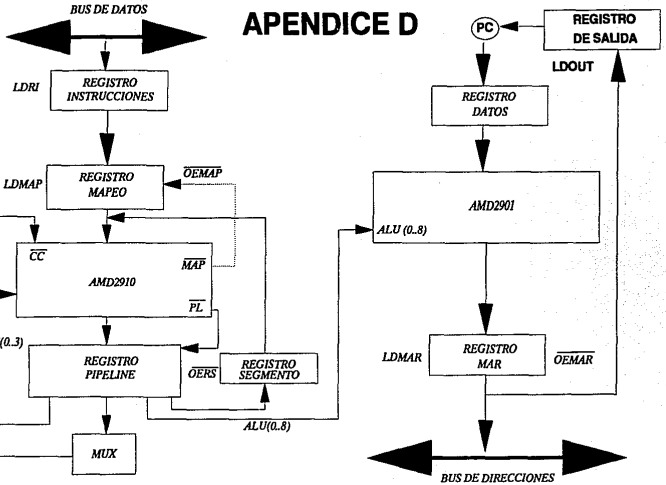
MNEMONICO	MICROCOFIGO				FUNCION RAM		FUNCION REG-Q		SALIDA Y	CORRIMIENTO RAM		CORRIMIENTO Q	
	I8	I7	I6	CODIGO OCTAL	CORRIMIENTO	CARGA	CORRIMIENTO	CARGA		RAM0	RAM3	Q0	Q3
QREG	L	L	L	0	X	NO	NO	F>Q	F	X	X	X	X
NOP	L	L	H	1	X	NO	X	NO	F	X	X	X	X
RAMA	L	H	L	2	NO	F>B	X	NO	A	X	X	X	X
RAMF	L	H	H	3	NO	F>B	X	NO	F	X	X	X	X
RAMQD	H	L	L	4	DOWN	F/2>B	DOWN	Q/2>Q	F	F0	IN3	Q0	IN3
RAMD	H	L	H	5	DOWN	F/2>B	X	NO	F	F0	IN3	Q0	X
RAMQU	H	H	L	6	UP	2F>B	UP	2Q>Q	F	IN0	F3	IN0	Q3
RAMU	H	H	H	7	UP	2F>B	X	NO	F	IN0	F3	X	Q3

FIG. 4 OPERADORES FUENTE Y MATRIZ DE FUNCIONES DEL ALU

		I 210 OCTAL							
		0	1	2	3	4	5	6	7
OCTAL I340	FUNCION ALU	FUENTE ALU							
		A,Q	A,B	0,Q	0,B	0,A	D,A	D,Q	D,0
0	Cn = L R PLUS S Cn = H	A + Q A + Q + 1	A + B A + B + 1	Q Q + 1	B B + 1	A A + 1	D + A D + A + 1	D + Q D + Q + 1	D D + 1
1	Cn = L S MINUS R Cn = H	Q - A - 1 Q - A	B - A - 1 B - A	Q - 1 Q	B - 1 B	A - 1 A	A - D - 1 A - D	Q - D - 1 Q - D	- D - 1 - D
2	Cn = L R MINUS S Cn = H	A - Q - 1 A - Q	A - B - 1 A - B	- Q - 1 - Q	- B - 1 - B	- A - 1 - A	D - A - 1 D - A	D - Q - 1 D - Q	D - 1 D
3	R OR S	A V Q	A V B	Q	B	- A	D V A	D V Q	D
4	R AND S	A & Q	A & B	0	0	A	D & A	D & Q	0
5	$\overline{R}$ AND S	$\overline{A}$ & Q	$\overline{A}$ & B	Q	B	0	$\overline{D}$ & A	$\overline{D}$ & Q	0
6	R EX-OR S	A V Q	$\overline{A}$ V B	Q	B	A	$\overline{D}$ V A	D V Q	D
7	R EX-NOR S	$\overline{A V Q}$	$\overline{A V B}$	Q	B	$\overline{A}$	$\overline{D V A}$	$\overline{D V Q}$	D

+ = PLUS; - = Minus; V = OR; & = AND; $\overline{V}$ = EX-OR

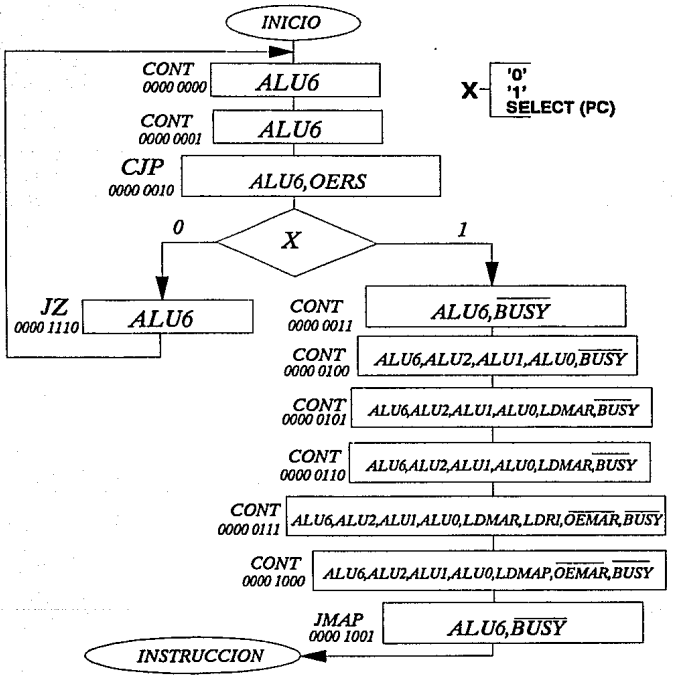
APENDICE D


BUS DE DATOS
LDRI
REGISTRO INSTRUCCIONES
LDMAP
REGISTRO MAPEO
OEMAP
CC
MAP
AMD2910
PL
I(0..3)
REGISTRO PIPELINE
OERS
REGISTRO SEGMENTO
MUX
ALU(0..8)
ALU (0..8)
AMD2901
PC
REGISTRO DE SALIDA
LDOUT
REGISTRO DATOS
LDMAR
REGISTRO MAR
OEMAR
BUS DE DIRECCIONES

No

Existe

Pagina



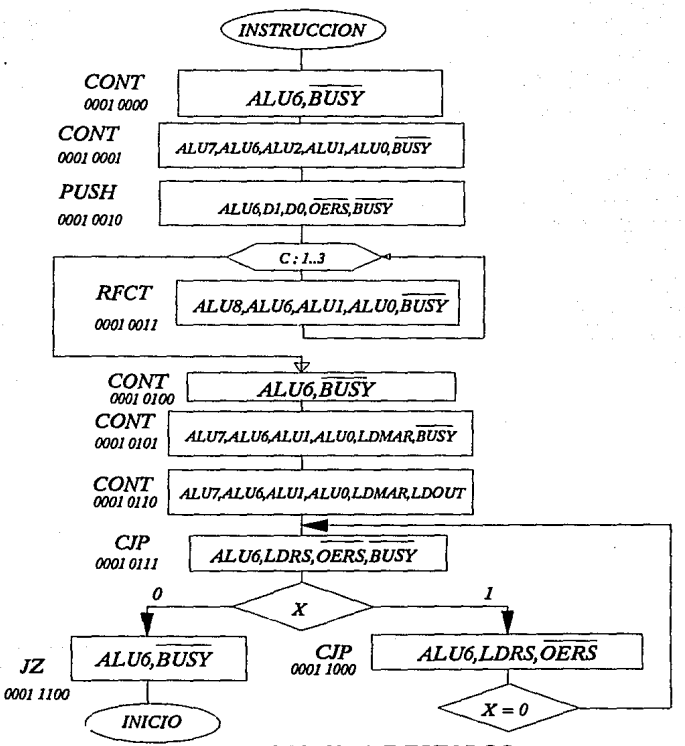
MAQUINA DE ESTADOS

APENDICE E

NO

Existe

Pagina



MAQUINA DE ESTADOS

APENDICE E

NO

Existe

Pagina

LISTADO DEL PROGRAMA PROTECTOR

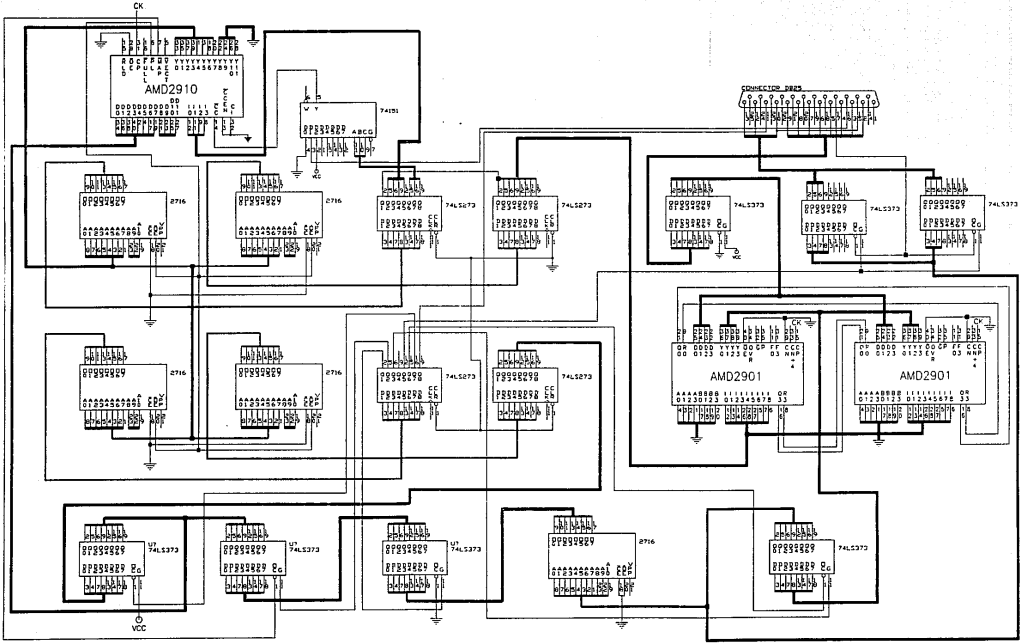
```
; Sistema de proteccion de informacion
;
; FACULTAD DE INGENIERIA / UNAM
; CESAR HUERTA OLIVARES / TESIS PROFESIONAL 1993
;
; OBJETIVO:
; - Este programa verifica si este existe residente en memoria;
;   si no esta activo, se instala, y si se encuentra activo se
;   desinstala.
;
; - Una vez residente, el programa detecta cualquier acceso a disco
;   y cambia la rutina del BIOS para el manejador de discos. La nueva
;   futina accesa el buffer de datos que maneja la interrupcion normal
;   INT 13H, ademas de comunicarse con el circuito ptorector externo que
;   contiene la rutina de encripcion.
;
; - Una vez manipulado el buffer del manejador de discos, se regresa el
;   control a la INT 13H del BIOS.
;
; ===============================================================

CSEG        SEGMENT
            ASSUME CS:CSEG
            ORG    100H
;
START:      JMP    INITIALIZE
;
;-----------------------------------------
; Definicion de variables de trabajo
;-----------------------------------------
;
OLDINT13    DD     ?              ; Originalmente vector de interrupcion 13
SWITCH      DB     0FH            ; ON/OFF switch de proteccion
ENC         DB     ?
CONTADOR    DW     ?
INSTR       DB     10H
DATO        DB     ?
SLC_ON      DB     07H            ; Bit de select = "1"
SLC_OFF     DB     0FH            ; Bit de select = "0"
LD_LOW      DB     00H            ; Bit init = "0" y Autofeed = "1"
LD_HIGH     DB     07H            ; Bit init = "1" y Autofeed = "0"
DATO_LOW    DB     ?
DATO_HIGH   DB     ?
;
; ------------------------------------
; NEW INTERRUPT 13 (BIOS DISK I/O)
; ------------------------------------
;
NEWINT13    PROC   FAR
            CMP    AH,03H         ; Se quiere escribir en disco ?
            JZ     CHECKSTAT      ; Si, checa el No. de drive
;
            CMP    AH,05H         ; Se quiere formatear un disco ?
            JZ     CHECKSTAT      ; Si, checa el No. de drive
;
CONTINUE:   JMP    CS:[OLDINT13]  ; Continuar con interrupcion normal
;
CHECKSTAT:  CMP    SWITCH,00H     ; Switch de proteccion prendido ?
            JNZ    CONTINUE       ; No, continua
            CMP    DL,02H         ; Se selecciono el disco 'C' ?
            JZ     CONTINUE       ; Si, continua
;
;-------------------------------
; Inicio : Rutina de encripcion
;-------------------------------
COMIENZO:
            PUSH   AX                    ; Se salvan los registros de trabajo
            PUSH   BX
            PUSH   CX
            PUSH   DX
;
            PUSH   SI
            PUSH   DI
;
            PUSH   DS
            PUSH   SS
            PUSH   ES
;
            PUSH   BP
            PUSHF
;
            CMP    CH,0            ;Evita area de BOOT y la FAT table
            JNZ    OK_ENC          ;Checa si el buffer tiene estas areas
            CMP    DH,0
```

```
            JE      OK_FIN
            CMP     CL,6
            JAE     OK_ENC
OK_FIN:     JMP     TERMINA             ;Si, entonces las copia en forma normal
OK_ENC:
            PUSH    CX
            PUSH    DX
            MOV     AH,0
            MOV     CX,512              ;Se calcula el numero de bytes a copiar
            MUL     CX
            MOV     CONTADOR,AX
            POP     DX
            POP     CX
;
REPITE:     MOV     AL,ES:BYTE PTR [BX]
            MOV     DL,AL               ; DL = dato
            CALL    CIRCUITO            ; LLama a la rutina de encripcion
            MOV     DL,DATO             ; externa.
            MOV     ES:BYTE PTR [BX],DL
            INC     BX                  ;Se posiciona en el siguiente dato
            DEC     CONTADOR
            CMP     CONTADOR,0          ;Se vacio el buffer ?
            JZ      TERMINA             ;Si, termina
            JMP     REPITE              ;No, lee siguiente dato del buffer
TERMINA:
;
            POPF                        ;Restaura registros de trabajo
            POP     BP
;
            POP     ES
            POP     SS
            POP     DS
;
            POP     DI
            POP     SI
;
            POP     DX
            POP     CX
            POP     BX
            POP     AX
            JMP     CONTINUE    ;Regresa el control a la rutina de
                                ;interrupcion original.
;
;
NEWINT13    ENDP
;
;
;-----------------------------------------------
; Esta rutina se comunica con el circuito protector
; externo. Se llama desde la rutina de encripcion.
;-----------------------------------------------
;
CIRCUITO :
        PUSH        AX
        PUSH        DX          ; Guarda los registros que se usan en esta rutina
;
        MOV         DATO,DL
;
        MOV         DX,379H     ; Direccion del puerto de entrada
WAIT:   IN          AL,DX       ; Lee dato del puerto
        TEST        AL,80H      ; Checa senal de busy
        JNZ         WAIT        ; Es "1" ? Esperar, si es "0" continuar
;
        MOV         AL,INSTR    ; Manda instruccion para el circuito externo
        MOV         DX,378H
        OUT         DX,AL
;
        MOV         AL,SLC_ON ; Indica que mando instruccion al circuito
        MOV         DX,37AH
        OUT         DX,AL
;
        MOV         DX,379H     ; Direccion del puerto de entrada
WAIT1:  IN          AL,DX       ; Lee dato del puerto
        TEST        AL,80H      ; Checa senal de busy
        JZ          WAIT1       ; Es "0" ? Esperar, si es "1" continuar
;
        MOV         AL,DATO     ; Manda instruccion para el circuito externo
        MOV         DX,378H
        OUT         DX,AL
;
        MOV         DX,379H     ; Direccion del puerto de entrada
WAIT2:  IN          AL,DX       ; Lee dato del puerto
        TEST        AL,80H      ; Checa senal de busy
        JNZ         WAIT2       ; Es "1" ? Esperar, si es "0" continuar
```

```
;
        MOV         AL,LD_LOW ; Direcciona dato menos significativo
        MOV         DX,37AH
        OUT         DX,AL
;
        MOV         DX,379H   ; Direccion del puerto de entrada
        IN          AL,DX     ; Lee dato del puerto
;
        ROL         AL,1      ; Ajusta el dato de entrada
;
        MOV         DATO_LOW,AL
;
        MOV         AL,LD_HIGH ; Direcciona dato mas significativo
        MOV         DX,37AH    ; en el bus de datos del circuito.
        OUT         DX,AL
;
        MOV         DX,379H   ; Direccion del puerto de entrada
        IN          AL,DX     ; Lee dato del puerto
;
        ROL         AL,1
;
        MOV         DATO_HIGH,AL

;
        MOV         AL,SLC_OFF ; Indica fin de instruccion
        MOV         DX,37AH
        OUT         DX,AL
;
        MOV         DX,379H   ; Direccion del puerto de entrada
WAIT3:  IN          AL,DX     ; Lee dato del puerto
        TEST        AL,80H    ; Checa senal de busy
        JZ          WAIT3     ; Es "0" ? Esperar, si es "1" continuar
;
        MOV         DL,DATO_LOW
;
        PUSH        CX        ;
        MOV         CL,4
        ROR         DL,CL
        POP         CX
        AND         DL,0FH
;
        MOV         AH,DATO_HIGH
        AND         AH,0F0H
;
        OR          DL,AH
        MOV         DATO,DL

        POP         DX        ; Restaura registros y regresa a donde se llamo
        POP         AX
        RET
;
; ------------------------------------------------
; Instalacion de programa
; Empieza la busqueda de existencia de copia de codigo
; ------------------------------------------------

INITIALIZE:MOV      DX,OFFSET NEWINT13 ; Offset de inicio de codigo de busqueda
           MOV      AX,CS     ; DS:SI apuntador a destino
           MOV      ES,AX     ; ES:DI apuntador a fuente
;
NEXTSEG:   DEC      AX        ; Busca segmento previo
           MOV      DS,AX     ; Carga nuevo segmento a buscar
           MOV      SI,DX     ; Apuntador al inicio de string
           MOV      DI,DX     ; Apuntador al inicio de string
; 4 palabras deben compararse para confirmar si existe copia del programa
           MOV      CX,0004H  ; Cuatro palabras deben compararse
           CLD               ; Borra DF para autoincrementar
           REPE     CMPSW
           JNZ      NOTFOUND  ; Si no se han comparado, sigue intentando
; Una copia local del programa puede existir en el buffer de entrada
; cualquier copia local debe ser identificada prendiendo el switch 0FH
           CMP      DS:SWITCH,0FH ; Es esta una copia instalada
           JNZ      TOGGLESW  ; Si es copia real, invierte el switch
NOTFOUND:  CMP      AX,0001H  ; Para de busar en localidades bajas de memoria
           JNZ      NEXTSEG
;
; Si salimos de un ciclo sin encontrar el codigo, entonces debemos
; instalarlo
;
           MOV      SWITCH,00H ; Prende el switch
           MOV      AX,3513H   ; Inicia la obtencion del viejo vector
           INT      21H
;
```

```
; Coloca el antiguo vector en memoria
            MOV     WORD PTR CS:[OLDINT13], BX
            MOV     WORD PTR CS:[OLDINT13+2], ES
;
            PUSH    CS
            POP     DS          ; Coloca DS en CS
;
; Envia mensaje, sale y permanece residente
            MOV     DX,OFFSET PROTECT_ON
            MOV     AH,09H      ; Imprime string de la llamada de la funcion
            INT     21H
;
; Hace la interrupcion 13 a este programa
            MOV     DX,OFFSET NEWINT13
            MOV     AX,2513H    ; Coloca la nueva interrupcion 13
            INT     21H
;
            MOV     DX,OFFSET INITIALIZE ; No. de bytes a guardar
            INT     27H         ; Termina y permanece residente
;
; Si existen programas en memoria, entonces debemos invertir unicamente el switc
TOGGLESW:   NOT     DS:SWITCH ; DS a sido prendido para busqueda
            CMP     DS:SWITCH,00H ; Esta el switch prendido
            JZ      ON
            MOV     DX,OFFSET PROTECT_OFF
            JMP     EXIT
ON:         MOV     DX,OFFSET PROTECT_ON
EXIT:       MOV     AH,09H      ; Imprime el string de la llamada de la funcion
            PUSH    CS
            POP     DS          ; Reinicia el DOS
            INT     21H
            INT     20H         ; Sale del DOS
;
PROTECT_ON  DB "PROTECCION DE DISCO ACTIVADA$"
PROTECT_OFF DB "PROTECCION DE DISCO DESACTIVADA$"
CSEG        ENDS
;
            END     START
```

CIRCUITO PROTECTOR DE SOFTWARE.



APENDICE G

DISEÑO : CESAR HUERTA OLIVARES.
TESIS PROFESIONAL 1993.
INGENIERIA EN COMPUTACION.

BIBLIOGRAFIA

Douglas V. Hall :Microprocessors and Digital Systems, Mc Graw-Hill International Editions, Singapore, 1987.

H. Taub., D. Schilling : Digital Integrated Electronics, McGraw-Hill International Editions, Singapore, 1986.

Douglas V. Hall :Microprocessors and Interfacing (Programming and Hardware), McGraw-Hill International Editions, Singapore, 1986.

M. Morris Mano :Digital Logic and Computer Design, Prentice-Hall, Inc, Englewood N.J., 1979.

M. Morris Mano :"Computer system architecture," 2ed.,Prentice-Hall, Englewood Cliffs,N.J., 1982.

R.H. Courney Jr.,: A Comparison of Commercial and Military Computer Security Policies, IEEE, Oakland, CA 1987.

J. Lobel, : Foling The System Breakers, MCGraw-Hill, INc., New York, NW 1986.

Meyer, C., y S. Matyas : Cryptography, A New Dimension in Computer Data Security, John Wiley & Sons, New York, 1982.

Knuth, D. The Art of Computer Programming, Vol. 2: Seminumerical Algorithms. Reading, MA: Addison-Wesley, 1969.

Dror, A.: Data Protection and Encryption, The Waite Group's MS-DOS papers. Howard W. Sams, 1988.

Wiatrowsky C., House C. : Logic Circuits and Microcomputer Systems, McGraw-Hill International Student Editions, Kogakusha, 1982.

Hamacher V.,Vranesic Z. : Computer Organization, McGraw-Hill International Student Editions, Singapore, 1984.

I.B.M. Corp., : Technical Reference, I.B.M. Personal Computer Hardware Reference Library, United Kingdom, 1984.

Duncan R., : Advanced MS-DOS, Microsoft Press MS-DOS Programming Reference, Redmon, WA 1986.

MS-DOS., : Technical Reference Encyclopedia ver (1.0-3.2), Microsoft Press, Redmon, WA 1986.

Chirlian, P.M.: "Analysis and Design of Digital Circuits and Computer Systems," Matrix, Champaign, IL,1976.

Hayes, J.P.: "Computer Architecture and Organization," 2d ed., Mc Graw-Hill, New York, 1984.

Hill, F.J., and G. R. Peterson: "Digital Systems:Hardwarw Organization and Design," 2d ed., Wiley, New York, 1978.

Poppelbaum, W.J.: "Computer Hardware Theory," Mc Graw-Hill, New York, 1972.Revistas :

Security Strategies., :Hardware Protection For PCs., PC Magazine, pp.105-120., April 28, 1987.

Security., : How safe is it?., BYTE Magazine, pp.254-291., June 1989.

Seguridad de la Información en la Red. : La Red en Peligro., Revista RED, Año II, Num. 16.

Encrypted ID's for digital Privacy. : Achieving Electronic Privacy., Scientific American Magazine, pp.76-81., August 1992, vol. 267, Num. 2.