DIVISION DE CIENCIAS BASICAS E INGENIERIA.

MAESTRIA EN CIENCIAS DE LA COMPUTACION.

TRIMESTRE 05-P

PROPUESTA DE PROYECTO DE TESIS.

“Arquitectura de seguridad y auditoría para un sistema de voto electrónico”

ALUMNO: Josué Figueroa González.
ASESORA: Dr. Silvia González Brambila.

I N D I C E .

1. I N T R O D U C C I O N .

Los avances tecnológicos y de computación han influido notoriamente en la manera en que se realizan muchas de las actividades actuales substituyendo a las formas tradicionales. Durante años los sistemas de votación han sido llevados a cabo mediante métodos convencionales como las boletas de papel, llamadas telefónicas o cartas y gran parte de los resultados se obtienen a través de un conteo manual. Con el paso del tiempo y el avance tecnológico surgieron los sistemas de voto electrónico, que comenzaron a ser utilizados en 1974, y eran conocidos como *DRE (Direct Recording Voting Systems)* o sistemas de almacenamiento directo del voto, originalmente eran referidos como un "contador electrónico de votos" [Saltman, 2003].

La computación abre una puerta que puede simplificar en muchos sentidos la tarea de realizar elecciones [Arango & Sánchez, 2004], de ahí que la computadora esté sirviendo de base fundamental para la implementación de votación electrónica, de esta manera, el voto electrónico es una realidad desde hace varios años en países como Francia, Alemania, Inglaterra, España, India, Brasil, Estados Unidos, Argentina y Venezuela entre otros, y se espera que dentro de poco este tipo de sistemas se aplique en México [FEPADE, 2004].

Sin embargo, dado el gran desconocimiento que existe por una parte muy grande de la población sobre este tipo de sistemas, se introduce también una gran desconfianza en muchos sentidos [Arango & Sánchez, 2004].

Las características que estos sistemas deben cumplir para que los usuarios tengan una buena confianza en ellos son: la opción registrada sea la misma que ellos quisieron emitir, los resultados finales deben mostrar realmente el sentir de todos aquellos que participaron en la elección, la opción que eligieron debe ser secreta, tener la seguridad de que los votos no pueden ser alterados y que el sistema sea sencillo de utilizar.

Es por eso que los sistemas de voto electrónico deben estar protegidos no solo contra acciones impropias por parte de los votantes, sino que también de las que pudieran

producirse por parte de los programadores, técnicos y administradores del sistema [Jones, 2004]. Además del aspecto de seguridad, estos sistemas deben incluir en su diseño conceptos como el de auditoría y auditabilidad que están relacionados con la confianza [Saltman, 2001].

En este documento se realiza un propuesta para diseñar una arquitectura que permita construir sistemas de voto electrónico presencial seguros y auditables, aunque varios de los aspectos a tratar pueden implementarse en un sistema de voto electrónico remoto ya que comparten muchas características.

2. A N T E C E D E N T E S.

En esta parte se presentan los conceptos básicos que se encuentran tanto en la auditoría como en la seguridad de este tipo de sistemas, se revisarán algunas de las opiniones o recomendaciones que han dado varios autores sobre estos temas, tanto desde el punto de vista de la auditoría como en el de la seguridad.

Se revisarán aspectos relacionados con la criptografía y su uso en los sistemas de voto electrónico, igualmente se revisará el tema de las vulnerabilidades que se presentan en estos sistemas tanto en el enfoque de auditoría como en el de seguridad.

Finalmente se revisarán dos sistemas de voto electrónico donde se describirá brevemente su funcionamiento y se realizará una crítica sobre las fallas que pudieran encontrarse al momento de su análisis.

2.1 AUDITORÍA.

2.1.1 Conceptos involucrados.

En el proceso de auditoría de un sistema de voto electrónico o de conteo, se pueden encontrar los siguientes elementos básicos: [Saltman, 2001]

- Auditoría.
- Auditabilidad.
- Confianza del votante.
- Confianza del público en general.

Auditoría.

Realizar una auditoría de un sistema basado en el conteo es examinar el sistema para determinar si los resultados que este sistema reporta son adecuados de acuerdo a la entrada y a las acciones del proceso. Una auditoría en este caso, servirá para corroborar que el total de votos que registró el sistema es congruente con el total de votos que recibieron en

conjunto los participantes de la elección y con el total de votos que emitieron los votantes [Saltman, 2001].

Auditabilidad.

Es determinar cuáles de los datos en los que se basará la auditoría están disponibles para ser usados y poder obtener un reporte adecuado. Se presenta un problema en la auditabilidad cuando los datos no se encuentran disponibles o el obtenerlos resulta muy difícil o costoso [Saltman, 2001].

Confianza del votante.

Se refiere a la certeza que tiene un votante de que su elección ha sido bien interpretada por la computadora. Además el votante quiere estar seguro de que su voto ha sido sumado correctamente a los votos que emitieron los demás votantes [Saltman, 2001].

Confianza del público en general.

Esto es el nivel de aceptación que tiene el público en general, tomando como un hecho, que los resultados representan las verdaderas elecciones de los votantes. Este nivel de confianza incluye la suma de las confianzas de los votantes, además de otros factores como el desempeño de los programas computacionales, los resultados de una auditoría y los reportes de la elección [Saltman, 2001].

2.1.2 Requisitos de los sistemas de voto electrónico auditables.

En lo que se refiere a la auditoría, estos sistemas deben respetar los siguientes requisitos: [Prince, 2004]

1 -Detección y registro permanente de cualquier evento significativo ocurrido con el dispositivo.

2 -Existencia de un reloj, que debe ser utilizado en todos los registros de eventos.

3 -El registro de eventos debe estar protegido contra fallas de energía; el dispositivo debe ser capaz de producir una versión impresa del registro de eventos.

Además este tipo de sistemas debe contar con las siguientes características: [Prince, 2004]

Confiabilidad: Los sistemas de votación deben funcionar de manera robusta, sin pérdida de votos ni de datos o información. Cabe destacar que en el voto electrónico la confiabilidad se basa fundamentalmente en una cuestión de percepción por parte de los electores y no tanto en una razón técnica.

Facilidad de uso: Se trata de diseñar métodos de votación fácilmente utilizables por los electores para que no generen confusiones en el elector ni en las autoridades encargadas del escrutinio.

Exactitud y posibilidad de verificación: Los sistemas de votación deben procurar el correcto almacenamiento de los votos y toda la información que registren. Y, en este sentido, todo el proceso debe poder ser verificable.

Exactitud: Para permitir un recuento perfecto.

Democraticidad: En la cual un elector es igual a un voto.

Privacidad: Nadie debe poder relacionar el voto con el votante.

Verificalidad: Que se traduce con la comprobación del correcto recuento.

Anonimato: Nadie puede saber lo que votó un elector.

Autenticidad: Sólo admitir a los votantes registrados

Singularidad: Ningún votante puede votar más de una vez.

Auditabilidad: Medios para verificar los resultados.

No coacción: Los votantes no pueden demostrar a otros por quién votaron.

Verificación individual: El votante debe poder comprobar su voto.

Los sistemas de voto electrónico deben enfocarse en los siguientes aspectos: [Selker, 2004]

a) Asegurar el almacenamiento del voto emitido, esto se logra con una arquitectura que logre la detección y la corrección de errores.

b) Prevención de alteraciones externas, especialmente las que involucren el cambio de los votos.

c) Prevención de alteraciones internas que incluyen el desarrollo malicioso de sistemas de voto electrónico.

d) Falsificación de los contenidos de los archivos que el sistema utiliza o genera.

2.1.3 Objetivos de una auditoría.

El principal objetivo de una auditoría es detectar errores o fraudes [Jones, 2004]. Una auditoría tiene al menos dos propósitos. El primero es determinar si la ley y los reglamentos se han seguido paso a paso en el proceso de votación y proporcionar evidencia de los errores que pudieron haber ocurrido, ya sea de manera accidental o de manera deliberada. El segundo es el de contribuir al aumento de confianza en las instituciones en caso de que la auditoría demuestre que los resultados son correctos [Saltman, 2001]. Como se puede observar, el propósito de una auditoría está muy relacionado con el concepto de la confianza del público en general, y este concepto no se aplica solamente a un sistema de voto electrónico sino también a otros aspectos en donde se utilicen aplicaciones informáticas.

2.1.4 Requerimientos para una auditoría.

El requerimiento principal para que un sistema sea auditable, es que tenga la suficiente información para permitir la detección y corrección de errores o falsificaciones. En este caso, se puede pensar que los sistemas auditables comparten varias características con los sistemas tolerantes a fallos. Otro requerimiento es que la persona u organización que realice las auditorías debe ser completamente ajena a los intereses de los grupos que están contendiendo en el proceso de elecciones. [Jones, 2004]

Como se ha mencionado se debe tener suficiente información de los resultados que arroja el sistema que se desee auditar, esto involucra el concepto de redundancia. La redundancia se considera uno de los principales aspectos para poder tener una auditoría eficiente [Jones, 2004].

El concepto de redundancia es fundamental, habilita al sistema para poder continuar trabajando de manera eficiente aún si existe una falla en algún instante dado [Selker, 2004]. Tener varios programas que procesen la misma información, incluso si es de manera diferente proporciona una buena medida de seguridad en el funcionamiento del sistema, lo que se traduce en un incremento de la confianza del público en el sistema [Selker, 2004].

2.1.5 Como realizar una auditoría.

La forma más básica pero que es muy eficiente para mostrar los resultados arrojados por una auditoría a un sistema de votación es la siguiente: [Jones, 2004]

$$V = C + U$$

Donde:

V : Votos almacenados

C: Suma de los votos registrados por cada candidato en la elección.

U: Número de votos nulos.

Todas estas cantidades deben ser positivas.

Esta igualdad se debe de cumplir, si no es así, la diferencia provee una medida del error en el conteo. Esta es quizá la forma más básica de auditar un sistema, pero va cobrando importancia si se realiza de forma continua durante el proceso de votación y no solo al final, ya que sirve para protegerse en contra de los ataques más obvios en el momento en el que se realiza la elección. Esto da origen a un sistema auto auditable. [Jones, 2004].

Una auditoría no debe estar basada en el recuento de los votos [Saltman, 2001], debe basarse en varios elementos que se deben proponer desde el mismo diseño del sistema, es por eso que es importante realizar un diseño pensando en obtener un sistema auditable [Jones, 2004].

2.1.6 Diseño de un sistema de voto electrónico auditable.

Los tres criterios que se deben tener en cuenta al momento de diseñar estos sistemas de votación son: [Figueroa et al., 2004].

- Debe ser sencillo para el votante emitir su voto.
- El sistema debe procesar correctamente la opción elegida.
- Debe existir confianza del público en los resultados que arrojará el sistema.

Propuesta de un sistema auditable de Roy G. Saltman.

En [Saltman, 2003] se proponen varias recomendaciones para diseñar un sistema, con el objetivo de que el sistema dependa lo menos posible de los factores humanos que intervendrán al momento de la votación y con la intención de generar un nivel de confianza mayor del votante y posteriormente del público en general.

(1) Deshabilitación del equipo después de que se ha completado el proceso de votación: Cada vez que se termine la etapa de votación para cada votante individual, el equipo se deshabilitará y no podrá recibir ningún otro voto hasta que sea habilitado nuevamente por alguno de los administradores del proceso de votación.

(2) Reporte directo hacia el votante de la contribución de su voto: Una prueba que ha demostrado generar un buen nivel de confianza es un pequeño contador visible para los votantes, que se incrementa en uno después de que han votado, a pesar de que no garantiza que el voto se le otorgó al candidato de su elección, el hecho de que el número de votos se incremente en uno parece producir una mejora en el nivel de confianza.

(3) Almacenar los votos y contabilizarlos en otra máquina: Es importante que los votos se almacenen en algún medio físico que pueda insertarse en otro dispositivo además de que se almacenen en la memoria interna de la urna, que servirá como respaldo de los resultados.

(4) Suma de los votos y de los votos nulos: Una buena prueba para una auditoría, es la suma de los votos, tanto los votos que registró cada candidato como los votos nulos, estos pueden manejarse por separado y después sumarse, o ir incrementando algún contador cada que se vote, independientemente del tipo de voto.

(5) Asegurar que todos los tipos de elecciones se le presenten al votante, puede utilizarse un indicador que le avise al votante si le falta votar en alguna de las elecciones o el sistema puede ir presentando todas las votaciones en un orden secuencial.

(6) Facilidad de pruebas: El sistema debe ser fácil de probar por parte de las autoridades encargadas de los procesos de votación, estas pruebas deben realizarse en presencia de los participantes interesados, estas pruebas no deberán afectar el futuro desempeño del sistema.

2.2 SEGURIDAD.

La seguridad de un sistema siempre es una mezcla de prevención, detección y respuesta. La prevención es hacer un blanco difícil o poco atractivo de atacar, la detección involucra identificar si se realizó o se está realizando un ataque y finalmente la respuesta que permite reaccionar al ataque detectado de manera decisiva para prevenir o disminuir sus efectos [García et al., 2004].

2.2.1 Criptografía.

El uso de la criptografía en estos sistemas proporciona un nivel más elevado de algunas de las propiedades que se cubren con la auditoría, especialmente en el aspecto de privacidad [Fischer, 2003]. Aunque la criptografía es sólo una parte pequeña de la seguridad de un sistema también se considera como una parte crítica, que permite que algunos tengan acceso a la información y otros no [Boneh, 1999]. La criptografía no es un problema, existen una gran cantidad de protocolos de criptografía que han sido probados satisfactoriamente, el problema principal es la arquitectura de seguridad que debe tener este tipo de sistemas, ya que según el principio de Kerckhoffs la seguridad de un sistema depende solo de la secrecía de la llave y no de la secrecía de los algoritmos [Bonhe, 1999].

Actualmente se emplean dos tipos de criptografía, la simétrica y la asimétrica.

Criptografía Simétrica.

En este tipo de criptografía se utiliza la misma contraseña o llave para encriptar y desencriptar la información. Entre algunos métodos de criptografía simétrica se pueden mencionar Blowfish, IDEA (*International Data Encryption Algorithm*), FEAL (*Fast Data Encipherment Algorithm*), DES (*Data Encryption Standard*) y los más comunes que son el 3-DES, y el Rijndael-AES, adoptado en 2000.

El usar la misma llave para encriptar y para desencriptar es un problema a la hora de enviar datos, ya que el remitente debe enviar previamente la llave al destinatario para que éste pueda desencriptar la información, y debe hacerlo por un canal seguro. Por lo tanto la criptografía simétrica se emplea especialmente para almacenamiento seguro de datos

(solamente una persona necesita la llave). Para envío de datos es preferible la criptografía asimétrica.

Criptografía Asimétrica.

Aquí se utilizan dos contraseñas o llaves, una llamada llave pública y una llamada llave privada. La información se encripta con la llave pública, y se desencripta con la llave privada. No presenta el problema de transmisión de la llave que tiene la criptografía simétrica, ya que la llave pública no sirve para desencriptar la información.

Los sistemas de criptografía asimétrica incluyen el DH (Diffie & Hellman), el ElGamal, el DSA (*Digital Signature Algorithm*), el Merkle-Hellman, el Chor-Rivest, el LUC, el McEliece, y finalmente el RSA (Rivest, Shamir & Adleman) que es el más ampliamente utilizado.

La criptografía asimétrica ofrece varias ventajas sobre la simétrica, como son: [García et al., 2004]

- No se requiere compartir llaves.
- Da origen al concepto de firmas digitales.
- Permite el establecimiento de identidad.

2.2.2 Seguridad del sistema.

Existe un problema fundamental que se debe solucionar cuando se diseñan sistemas de voto electrónico, el "problema de la plataforma segura" [Rivest, 2001]. En cuanto a los problemas de seguridad de este tipo de sistemas se pueden resumir en tres aspectos fundamentales [Kohno et al., 2004]:

- Seguridad de los votos almacenados.
- Seguridad de los datos críticos.
- Seguridad del código fuente.

2.2.2.1 Seguridad de los votos almacenados.

Los aspectos ideales para un sistema de voto electrónico en lo que se refiere a los votos almacenados en el mismo son [Cranor & Cytron, 1997]:

- Correctez.
- Invulnerabilidad.
- Privacidad.

Correctez.

Un sistema es correcto si no es posible alterar un voto, si no es posible para un voto válido el ser eliminado del conteo final y si no es posible contabilizar un voto no válido [Cranor & Cytron, 1997]. En la mayoría de los sistemas con esta propiedad el resultado final será adecuado debido a que no se generaron errores o si se generaron pudieron corregirse.

Invulnerabilidad.

Un sistema es invulnerable si permite que solo voten electores autorizados y que estos lo realicen solo una vez [Cranor & Cytron, 1997].

Privacidad.

Un sistema es privado si ningún tipo de autoridad o nadie más puede hacer una relación de un voto con la persona que lo emitió y que ningún votante puede probar que votó por alguien el particular [Cranor & Cytron, 1997]. Los votos deben ser almacenados de manera que no se pueda tener una relación con la persona que lo emitió, es por eso que se deben de almacenar de manera aleatoria.

2.2.2.2 Seguridad de los datos críticos.

En los sistemas de votación, proteger la integridad y la privacidad de los datos críticos es de suma importancia [Kohno, 2004]. Estos archivos se pueden cargar en el sistema de varias maneras, puede ser introduciendo un medio de almacenamiento como un disquete o una memoria, o descargando la información de alguna red. Es importante que estos archivos

tengan una forma de comprobar que provienen de donde deben y que no han sido modificados en el trayecto, para esto es importante el manejo de las firmas digitales.

Firma digital.

Es una forma de asegurar que el mensaje no ha sido modificado por terceros y que procede de la persona que asegura haberlo enviado, esta se puede utilizar cuando no se quiere ocultar ninguna información pero se quiere asegurar que no ha sido modificada.

Encriptado con firma.

Es una mezcla de la firma digital con algún método de encriptado, se deberá primero descifrar el mensaje y posteriormente se verificará que no ha sido alterado y que realmente proviene de quien asegura enviarlo.

En cuanto a los resultados generados y los archivos de eventos, estos también deben ser encriptados y firmados digitalmente, pero si esto se realiza de manera incorrecta de nada servirá. Un ejemplo de esto es el sistema creado por Diebold para las elecciones del año 2000 en Estados Unidos, aquí se utilizaba la misma llave para encriptar toda la información, pero el error más grave era que la clave se incluía en el mismo código fuente, se encontraba en lo que se conoce como "*hardcoded*" cuando estas claves deben ser generadas de manera aleatoria cada vez que son necesitadas [Kohno et al., 2004].

2.2.2.3 Seguridad en el código fuente.

Cuando se crea un sistema, obtener el diseño adecuado es solo una parte, el diseño debe ser implementado de manera correcta [Kohno et al., 2004].

El leer el código fuente de un programa da una visión no muy completa de las acciones que este realiza, se debe incluir comentarios muy detallados del funcionamiento de cada una de las distintas rutinas, funciones o módulos, de igual manera se deben proveer versiones anteriores del código y hacer énfasis en los cambios que se les han realizado [Kohno et al., 2004].

El código debe examinarse para evitar que incluya rutinas que alteren de manera intencionada o no el funcionamiento del programa, otra propiedad es que debe estar libre de rutinas que lo hagan caer en ciclos infinitos, que produzcan violaciones de segmentos o sobre flujos, que las variables se encuentren inicializadas de manera correcta y evitar en lo posible las definiciones *hardcoded* en especial si se trata de información relacionada con la seguridad.

Importancia del código abierto.

Algunos expertos proponen el uso de código abierto para los sistemas de votación [Seifert, 2002], este código debe estar disponible para la revisión pública, lo que lo hará más seguro, ya que al estar expuesto a una verificación por parte del público en general, será posible el identificar posibles fallas de seguridad de las que se detectan en los códigos de propiedad.

Mientras que la secrecía del código puede ser una medida de seguridad importante llamada "seguridad a través de la oscuridad", también posee varias debilidades. Primero es frágil ya que si se rompe esta defensa, el daño no puede repararse (no se puede hacer secreto el código de nuevo). Segundo, el uso de la secrecía limita el número de personas que pueden examinar el código y por tanto el escrutinio que ayudaría a detectar vulnerabilidades [Fischer, 2003].

2.3 VULNERABILIDAD.

El concepto de vulnerabilidad se encuentra presente tanto en el aspecto de auditoría como en el de seguridad.

En cuanto a la vulnerabilidad relacionada con la auditoría, el principal problema no tiene un enfoque técnico, más bien el problema es de confianza en el sistema. El principal problema que presentan muchos de estos sistemas es la falta de un comprobante que sirva al votante para verificar su voto [Saltman, 2003].

En cuanto a la seguridad, la principal vulnerabilidad está relacionada con el almacenamiento de los votos y los tres conceptos mencionados anteriormente, la correctez, la invulnerabilidad y la privacidad.

También se tiene vulnerabilidad en la configuración de estos sistemas, los archivos críticos deben de estar encriptados y deben contar con una manera de poder ser identificados y de poder verificar que no han sido modificados, es aquí donde cobra importancia el concepto de firma digital. Una de las principales vulnerabilidades de la configuración se da debido a la forma en la que se cargan los datos, ya que si se cargan de manera centralizada por medio de una computadora, se evita un posible error humano, pero se corre el riesgo de que se altere la información en esa computadora y se afecten todos los dispositivos [Saltman, 2001].

En cuanto a las vulnerabilidades del software, se pueden tener errores lógicos, la posibilidad de "código oculto" y de cambios al código no documentados.

2.4 ANALISIS DE SISTEMAS EXISTENTES.

El sistema de voto electrónico Diebold.

Su funcionamiento era en base a una pantalla táctil, era activado por medio de tarjetas inteligentes y los votos se iban almacenando en el sistema, cuando el proceso terminaba, estos se copiaban a un archivo el cuál se transmitía a un centro encargado de realizar el conteo.

Este sistema representó un gran problema en las elecciones presidenciales del 2000 en el estado de Florida, a continuación se presentan algunas de sus fallas [Kohno et al., 2004].

- El primer problema que se detectó fue en los archivos de configuración del sistema, estos no se encontraban protegidos ni por una técnica de encriptado o de verificación para detectar si en algún momento habrían sido modificados.

- En cuanto a la privacidad, el sistema almacenaba los votos de manera secuencial por lo que era muy sencillo ir relacionando el orden de los votos con el orden de los votantes. Otra versión del sistema almacenaba los votos de manera aleatoria, pero se cometió el error de asociar la posición en la que se guardaba el voto con un número de serie de la tarjeta inteligente utilizada para la activación del equipo, por lo que era fácil relacionar el voto con el votante. A pesar de que los votos se almacenaban encriptados, al finalizar la votación los resultados se copiaban a un archivo y se transmitían sin ninguna medida de seguridad que evitara su modificación o detección en caso de que estos fueran alterados.

- El código fuente no se encontraba completamente detallado, e incluso se tenían módulos sin ningún tipo de comentarios sobre su funcionamiento. El error más grave se detectó en la clave de encriptado de los archivos generados por el sistema ya que esta se encontraba con una definición "*hardcoded"* por lo que cualquiera que hubiera visto el código fuente podría conocer esta clave y modificar el contenido de cualquier archivo. Este código era cerrado, pero se rompió la defensa de “seguridad basada en la oscuridad” y se hizo público, lo que ocasionó que se detectaran las fallas antes mencionadas.

Sistema de voto electrónico “Tinochtin”.

Este sistema se construyó en la Universidad Autónoma Metropolitana unidad Azcapotzalco bajo el convenio de colaboración propuesto por el Instituto Electoral del Distrito Federal para la elaboración de un prototipo de urna electrónica para las elecciones en el Distrito Federal.

El sistema funcionaba en base a una pantalla táctil, que registraba la elección del votante, después presentaba una pantalla con la opción de corregir o confirmar la opción seleccionada, al confirmar el voto se generaba un comprobante impreso para que el votante pudiera verificar su elección. Los votos se almacenaban encriptados en un archivo binario de acceso aleatorio de manera aleatoria. Además de almacenarse en otros medios. Una vez terminado el proceso de votación se generaban actas con los resultados de la elección y los

resultados se encriptaban y copiaban a un medio de almacenamiento (memoria USB) para ser enviados posteriormente a un equipo que integraba los resultados de acuerdo a la sección electoral. La forma de activación que utilizaba era a través de códigos de barra generados de manera aleatoria uno para cada votante.

- El sistema contaba con un archivo de bitácora que registraba los principales eventos que ocurrían en el sistema.
- Este sistema tenía algunos problemas con la auditoría, ya que si bien generaba un comprobante impreso, no presentaba al votante una contribución directa de su voto, detectaba si se le habían restado votos a un partido, pero no lo hacía en caso de que se le hubieran agregado.
- En cuanto a la seguridad algo importante era que se encriptaba cada voto de manera individual con su propia clave, pero los votos almacenados en forma de archivo se encriptaban con la misma clave, igual ocurría con los archivos generados por el sistema, como actas y otros reportes. Existía un problema al momento de ordenar de manera aleatoria los votos, ya que en cada ejecución, estos quedaban en el mismo orden. Se contaba con un esquema de firmas digitales con encriptado aunque esto se realizaba con aplicaciones que requería de llamadas al sistema durante la ejecución del programa.

- El código fuente no se encontraba lo bastante detallado, aunque en este no se tenían definiciones "*hardcoded"* de ningún tipo.

3. JUSTIFICACION.

Uno de los principales problemas en los sistemas informáticos es el de la “plataforma segura” que es cuando no se cuenta con una arquitectura bien definida para diseñar un sistema seguro, este problema también afecta a los sistemas de voto electrónico los cuales además deben contar con una arquitectura de auditoría.

Actualmente no se cuenta con una arquitectura bien definida que ayude a diseñar de manera eficiente estos sistemas, lo que se tiene es una gran cantidad de opiniones y recomendaciones acerca de sus puntos críticos y lo que se realiza es ir agregando defensas una vez que el sistema ha sido vulnerado, cuando lo correcto es que este sea seguro y confiable desde el momento de su creación.

La importancia de este proyecto radica en identificar los aspectos vulnerables de los sistemas de voto electrónico desde el punto de vista de la seguridad y de la auditoría, y de acuerdo a esto diseñar una arquitectura que además de considerar los problemas que se tienen en estos sistemas proporcione una buena metodología para resolverlos.

La arquitectura obtenida podría servir como una base que contenga todos los requisitos que estos sistemas deben cubrir y además proporcione una manera de resolverlos, esta será de gran utilidad cuando se desee diseñar un sistema de voto electrónico seguro y auditable.

4. OBJETIVOS.

4.1 Objetivo general.

- Diseñar una arquitectura que permita construir sistemas de voto electrónico presencial seguros y auditables.

4.2 Objetivos específicos.

- Determinar las vulnerabilidades de los sistemas de voto electrónico presencial desde el punto de vista de la seguridad y la auditoría.
- Desarrollar una metodología para resolver de manera eficiente estas vulnerabilidades.
- Determinar los elementos necesarios para decir si un sistema de voto electrónico presencial es seguro y auditable.
- Implementar la arquitectura diseñada a un sistema de voto electrónico para determinar su eficiencia mediante el proceso de pruebas paralelas.

5. METODOLOGIA.

A continuación se describen algunas herramientas que se utilizarán para el desarrollo del proyecto, también se hablará de la técnica de diseño de la arquitectura y de su tipo, el primer concepto que se debe conocer es el de una arquitectura de seguridad, que es el proceso de seleccionar elementos y principios de diseño que cumplan con las necesidades de seguridad del sistema [Graff et al., 2003]. Para realizar el diseño de una arquitectura de software es importante considerar el ciclo de vida del desarrollo de software, que comienza con la identificación de una necesidad y termina con la verificación formal del software desarrollado en contra de esta misma [IPL, 1997].

Para el desarrollo de una arquitectura se tienen varios métodos, entre los que se encuentran el secuencial o de cascada y el progresivo o iterativo. El proceso secuencial se basa los siguientes pasos: Recopilación de requisitos, desarrollo, pruebas y entrega final. El método progresivo consta de las siguientes partes: Requisitos, diseño, implementación y pruebas y revisión, pero la a diferencia del secuencial, aquí se tiene un ciclo iterativo que permite ir creando versiones del sistema que si bien al principio no cumple con los requisitos establecidos si cuenta con una funcionalidad para realizar distintas pruebas y en base a ello modificar el diseño o la forma de implementación, un vez que las pruebas son satisfactorias esa versión sirve como base para aplicar nuevamente los pasos del proceso iterativo y así hasta que el sistema finalmente cumple con los requisitos deseados [Referencia]

Los requerimientos de auditoría y seguridad se obtendrán revisando el estado del arte con el fin de obtener la mayor cantidad de información acerca de los puntos que se deben cubrir para obtener un sistema de voto electrónico seguro y auditable, una vez reunidos estos requisitos se diseñará una manera de resolverlos y con eso se irá diseñando la arquitectura.

El aspecto de encriptado, se tratará con el paquete de aplicaciones y librerías OpenSSL, para la implementación de criptografía simétrica se realizará utilizando alguno de los algoritmos de encriptado que proporciona la Interfaz de Programación de Aplicaciones (API) de OpenSSL, esto gracias a la interfaz EVP contenida en este paquete. La elección

del algoritmo de encriptado dependerá de sus propiedades, como seguridad, rapidez, o incluso la libertad de implementación, para esta etapa se tienen dos opciones viables, los algoritmos de encriptado CAST5 e IDEA, esto es debido a que otros algoritmos disponibles cuentan con varias desventajas, por ejemplo el algoritmo *Blowfish* no es adecuado cuando se quieren utilizar varias llaves para encriptar una pequeña cantidad de datos, el DES que solo admite llaves de máximo 56 bits, lo que lo hace vulnerable a varios ataques, el RC2 que si bien no presenta vulnerabilidades, tampoco ha sido sometido a un escrutinio a fondo para estudiar su funcionamiento y el RC4 y RC5 que resultan difíciles de utilizar y requieren un permiso de los laboratorios RSA [Viega et al., 2002].

La implementación de los algoritmos de criptografía asimétrica se realizará con la interfaz EVP_PKEY, que permitirá la generación de llaves pública y privada que se utilizarán para las firmas digitales, en este caso OpenSSL cuenta con tres opciones: DH, DSA y RSA, para la implementación se utilizará el algoritmo RSA ya que es el único que cuenta con la capacidad de encriptar y firmar datos, DH solo es capaz de reconocimiento de llaves pero no puede realizar firmas digitales ni encriptar mientras que DSA puede realizar firmas digitales pero no es capaz de encriptar [Viega et al., 2002].

La construcción de la arquitectura se realizará de manera iterativa, lo que permitirá ir generando varias versiones de esta misma que poco a poco vayan cumpliendo con todos los requisitos que se hayan especificado. Esta podría realizarse de manera secuencial diseñándola por completo y luego implementándola, pero esto resultaría más complejo y se tendría una mayor probabilidad de cometer errores y el resolverlos implicaría el cambiar muchos aspectos del diseño. El diseño progresivo que se ha elegido proporciona varias ventajas, como el realizar pruebas de los componentes de la arquitectura en un sistema e ir avanzando de manera paralela tanto en el desarrollo como en la implementación, otra es que cuando se genera una nueva versión se hace tomando como base una ya probada, lo que facilita la detección de errores y finalmente cuando se tenga la versión definitiva se sabrá que ya funciona de manera eficiente sin la necesidad de tener que realizar todas las pruebas.

El tipo de arquitectura a utilizar se definirá durante la revisión del estado del arte examinando también los diferentes tipos de arquitecturas existentes, aunque como primera aproximación se piensa en una arquitectura de capas, en donde cada capa represente uno de los puntos por donde este tipo de sistemas puede ser atacado. La idea es proteger el núcleo del sistema que en este caso son los votos almacenados para evitar que puedan ser modificados, las capas se irán creando de acuerdo a las especificaciones de seguridad que se vayan recopilando.

Las pruebas a la arquitectura se realizarán implementando sus componentes al sistema de voto electrónico desarrollado por la UAM (Tinochtin) y aplicando el concepto de pruebas paralelas [Jones, 2004] que consiste en tomar el sistema y probarlo violando varios aspectos de auditoría. Estas pruebas se basarán en las vulnerabilidades de este tipo de sistemas y se verá que tan bien responde la arquitectura propuesta a este tipo de ataques, esto se hará tratando de vulnerar cada una de las capas hasta llegar a los votos almacenados para poder modificarlos, por el tipo de diseño de la arquitectura, las pruebas se realizarán cada vez que se desarrolle una aplicación para cubrir un requisito de seguridad o de auditoría por lo que solo se procederá al diseño e implementación de otro nivel cuando se tenga la seguridad de que el actual cumple con los requisitos deseados.

En cuanto al encriptado simétrico, los algoritmos propuestos ya han sido revisados por varios expertos [Adams, 1997], así como su eficiencia dependiendo del tamaño de llave utilizada, por lo que como prueba solo se estudiarían sus propiedades.

El algoritmo RSA implementado para llaves pública y privada, proporciona una alta seguridad ya que con una llave de longitud adecuada el proceso de descifrarla podría tardar hasta 1000 años con la tecnología actual. [García et al., 2004]

6. RECURSOS.

6.1 Recursos requeridos.

Computadora Pentium III a 750MHz, con 192Mbytes de memoria RAM, CD-ROM, puerto serial y puerto USB, una memoria USB, impresora térmica Fujitsu con puerto serial.

Sistema Operativo Linux SUSE Profesional 8.0, la versión más actualizada (0.9.7) que OpenSSL que se puede descargar de manera gratuita de: http://www.openssl.org/source/, compilador (g++) para lenguaje C++ también disponible en el sistema Linux.

La arquitectura se probará implementándola en el sistema de voto electrónico “Tinochtin” diseñado en la UAM unidad Azcapotzalco cuyo código fuente se encuentra disponible.

6.2 Recursos disponibles.

Todos los recursos que se necesitan se encuentran disponibles, tanto en hardware como en software.

7. C A L E N D A R I Z A C I O N.

Trimestre 1.

Actividad	Semana										
	1	2	3	4	5	6	7	8	9	10	11
Revisión del estado del arte.											
Organización y análisis de los requerimientos de auditoría.											
Organización y análisis de los requerimientos de seguridad.											
Diseño de los requerimientos de auditoría.											
Desarrollo de los requerimientos de auditoría.											
Implementación de los requerimientos de auditoría en el sistema.											
Estudio del API EVP.											
Selección del algoritmo de encriptado simétrico.											
Desarrollo del encriptado simétrico.											
Implementación del encriptado simétrico en el sistema.											
Pruebas de las distintas versiones del sistema.											
Construcción de la arquitectura.											
Elaboración del reporte.											
Elaboración de la presentación.											

Trimestre 2.

Actividad	Semana										
	1	2	3	4	5	6	7	8	9	10	11
Estudio del API EVP_PKEY.											
Desarrollo del encriptado asimétrico.											
Implementación del encriptado asimétrico en el sistema.											
Desarrollo del esquema de firmas digitales.											
Implementación del esquema de firmas digitales en el sistema.											
Pruebas de las distintas versiones del sistema.											
Construcción de la arquitectura.											
Desarrollo del reporte.											
Desarrollo de la presentación.											

8. REFERENCIAS.

- [Adams, 1997]
  ADAMS, C. *The CAST-128 Encryption Algorithm*, 1997.
- [Arango & Sanchez, 2004]
  ARANGO, R. SANCHEZ, F. *Voto electrónico*, Todo Linux No 30, 2004.
- [Boneh, 1999]
- BONEH, D. *Twenty Years of Attacks Against the RSA Crypto-system*, Notices of the American Mathematics Society, 5(2), 1999.
- [Cranor, 1997]
  CRANOR, L. CYTRON, R. *Secure Voting Systems*, Proceedings of the Hawai`s International Conference on System Sciences Hawaii U.S.A, 1997.
- [FEPADE, 2004]
  FEPADE, *La urna electrónica: Avances y Prospectivas*, Simposium sobre Urnas Electrónicas organizado por el IEDF realizado en México D.F, 2004.
- [Figueroa et al., 2004]
  FIGUEROA, J. GONZALEZ, S. ZARAGOZA, F. *Auditabilidad de urnas electrónicas*, Simposium sobre Urnas Electrónicas organizado por el IEDF realizado en México D.F, 2004.
- [Fischer, 2003]
  FISCHER, A. *Election Reform and Electronic Voting Systems (DREs): Analysis of security issues*, CRS Report for Congress, 2003.
- [García et al., 2004]
  GARCIA, L. MORALES, G. GONZALEZ, S. *Implementación del algoritmo RSA para su uso en el voto electrónico*, Simposium sobre Urnas Electrónicas organizado por el IEDF realizado en México D.F, 2004.
- [Graff, 2003]
  GRAFF, M. VAN W, K, *Secure Coding*, 2003.
- [IPL, 1997]
  IPL. *Software testing and software development lifecycles*, 1997.

- [Jones, 2004]

  JONES, D. *Auditing Elections*, Communications of the ACM Vol 47. No 10, pp 46-50, 2004.

- [Kohno et al., 2004]

  KOHNO, T. STUBBLEFIELD, A. RUBIN, A. WALLACE, D. *Analysis o fan Electronic Voting System*, IEEE Symposium on Security and Privacy, 2004.

- [Prince, 2004]

  PRINCE, A. *Consideraciones aportes y experiencias para el voto electrónico en Argentina*, Buenos Aires, 2004.

- [Rivest, 2001]

  RIVEST, L. *Electronic Voting*, 2001.

- [Seifert, 2002]

  SEIFERTT, J. PRIMER, A. *Computer Software and Open Source Issues*, CRS Report RL31627, 2002.

- [Selker, 2003]

  SELKER, T. GOLLER, J. *The SAVE System: Secure Architecture for Voting Electronically*, BT Technology Journal Vol 22 No 44, pp 89-95, 2003.

- [Saltman, 2003]

  SALTMAN, R. *Auditability of non-ballot, poll-site voting systems, Roy G. Saltman*, 2003.

- [Saltman, 2001]

  SALTMAN, R. *Auditability and Voter Confidence in Direct Recording (DRE) Voting Systems*, 2001

- THE LIBRARY OF CONGRESS. *Election Reform and Electronic Voting Systems (DREs): Analysis of Security Issues*, 2003.

- [Viega, 2002]

  VIEGA, J. MESSIER, M. CHADRA, P. *Network Security with OpenSSL*, 2002.