Esquema de Almacenamiento Seguro de Llaves Criptográficas

Roberto Gómez Cárdenas, Ricardo C. Lira Plaza, Adolfo Grego

Departamento de Ciencias Computacionales
Instituto Tecnológico y de Estudios Superiores de Monterrey – Campus Edo. de México
Apdo. Postal 6-3, Módulo Servicio Postal, Atizapán C.P. 52926, Edo. México
{rogomez, rlira, agrego}@campus.cem.itesm.mx

Resumen La seguridad computacional se define como el conjunto de políticas y mecanismos que nos permiten garantizar la confidencialidad, la integridad y la disponibilidad de los recursos de un sistemas. Hoy en día se utilizan diferentes criptosistemas para proporcionar seguridad a los sistemas, sin embargo dependen de una llave criptográfica. Nuestro trabajo propone un esquema que involucra la teoría de secretos compartidos en un ambiente distribuido que permite salvaguardar la integridad así como la confidencialidad de las llaves. Además nuestra propuesta incorpora tolerancia a fallas por lo que el aspecto de la disponibilidad también es considerado

Palabras clave: criptología, secretos compartidos, seguridad computacional, redes, sistemas distribuidos

1. Introducción

En la actualidad, la mayoría de las empresas dependen en gran medida de su información y una buena práctica para garantizar su confidencialidad ha sido el incorporar herramientas criptográficas. Sin embargo, no siempre se está consciente de lo que esto implica ya que se propociona seguridad a la información pero casi siempre se olvida la administración de la llave.

La base de nuestro esquema es aceptar la posibilidad de que nuestro sistema sea penetrado ya que nadie puede garantizar una seguridad perfecta. Lo anterior no implica que información sensitiva, como lo son las llaves criptográficas, quede en manos de algún intruso. Lo que proponemos es un esquema en el que dicha información no esté físicamente en el sistema sino distribuida, a través de secretos compartidos, a N equipos ubicados en una red que suponemos segura.

La criptografía nos permite incorporar tanto confidencialidad como integridad, autenticación, control de acceso y no repudio al transmitir y/o almacenar nuestra información. Sin embargo, no basta un criptosistema para garantizar la seguridad de la información, ya que dicha seguridad depende de la seguridad de nuestras llaves. Esta situación es ventajosa ya que en lugar de preocuparnos por la seguridad de una gran cantidad de información ahora sólo necesitamos salvaguardar una pequeña porción: las llaves criptográficas. Esto mismo nos hace reflexionar sobre la necesidad de una técnica no criptográfica, es decir, que no dependa de una contraseña, para brindar seguridad a nuestras llaves. Sería un grave error el almacenar en el mismo lugar físico tanto el texto cifrado como las llaves, por esto proponemos un ambiente distribuido que además incorpore tolerancia a fallas.

La administración de llaves involucra su generación, autenticación, almacenamiento y distribución. La seguridad de las llaves solo puede garantizarse con una adecuada administración. Por esto la seguridad que otorguemos a nuestras llaves deberá ser similar a la sensibilidad de la información que se este protegiendo.

La teoría de Secretos Compartidos desarrollado por Adi Shamir [1] es la base de nuestro esquema. Cuenta con características que la hacen muy adecuada para nuestra aplicación, como el no depender de una contraseña para brindar confidencialidad a la información; tolerancia a fallas a través de dividir la información en N partes y poder reconstruirla con M partes donde M<N.

Proponemos el crear un ambiente distribuido a través de la técnica de secretos compartidos para salvaguardar la integridad y brindar confidencialidad a las llaves criptográficas, tanto simétricas como

asimétricas. Es importante señalar que en este artículo abordamos solo el tema de salvaguarda las llaves, sin embargo, el esquema permite asegurar cualquier tipo de información sensible de un sistema expuesto a un ambiente hostil, como lo podrían ser las bitácoras, archivo de contraseñas, el contenido público de nuestro servidor de web, información de los usuarios del sistema, huellas de los distintos programas que se usen en el sistema (con lo cual podemos evitar caballos de troya y códigos maliciosos), bases de datos, así como documentos con información clasificada.

El trabajo se encuentra dividido de la siguiente forma: en la siguiente sección se presenta un panorama general de la administración de llaves. La sección tres explica los conceptos fundamentales de los secretos compartidos. La sección cuatro da a conocer nuestra propuesta, mientras que la sección cinco la implementación de ésta. Por último se presentan las conclusiones y el trabajo a futuro.

2. Las llaves criptográficas

La criptografía es la ciencia y el arte de escribir en clave, manteniendo la información en secreto [2]. El cifrado es el proceso por el cual cierta información se transforma en información cifrada a través de un algoritmo y una llave. El descifrado a su vez es el proceso por el cual recuperamos la información original al aplicar un algoritmo y una llave a la información cifrada. Así los elementos básicos de la criptografía son los algoritmos de cifrado y descifrado así como la llave. La llave determina el tipo de transformación que se realiza sobre los datos y elementos de información. En los sistemas computacionales e informáticos, la llave es una cadena de datos (almacenada electrónicamente).

Existen dos tipos de criptografía: simétrica y asimétrica. En un criptosistema de llave secreta, se emplea la misma llave para realizar los procesos de cifrado y descifrado. En un criptosistema de llave pública se emplean dos llaves criptográficas, una que realiza la función de cifrado y la segunda que realiza la función de descifrado (el procedimiento es muy semejante al que realizan los bancos con las cajas de seguridad de valores personales).

La criptografía nos permite proteger la información, asegurar su autenticidad y detectar modificaciones no autorizadas durante su almacenamiento y transmisión. La seguridad del proceso de cifrado depende de varios factores. Primero, el algoritmo de cifrado debe ser lo suficientemente robusto para que sea computacionalmente imposible descifrar un mensaje, con el simple hecho de conocer el mensaje cifrado. El algoritmo DES demostró por varios años ser un algoritmo con éstas características, hoy en día es RIJNDAEL el nuevo estándar de cifrado avanzado. Segundo, la confidencialidad de la información depende de la secrecía de la llave no del algoritmo, este último es público. Por esto también debe ser computacionalemente imposible descifrar un mensaje si conocemos tanto el algoritmo de cifrado como el texto cifrado.

En el caso de la criptografía asimétrica, por ejemplo RSA, Diffie-Hellman, ElGamal, DSA, se generan dos llaves, una pública y una privada. Como su nombre lo establece la primera es conocida por todo el mundo y la segunda se mantiene en secreto. Es esta llave la que buscamos proteger con nuestro esquema. Aquí también requerimos algoritmos tanto de cifrado como de descifrado robustos. Se utiliza la llave pública para cifrar información y su respectiva llave privada para descifrar. Esto trae como consecuencia que, en caso de que se comprometa la llave privada, toda la información que se ha cifrado con ésta también se verá comprometida.

Dada la importancia de la llave, es necesario un esquema de seguridad de llaves durante su almacenamiento que garantice que al ser penetrado un sistema no se comprometa la confidencialidad de la información. Como vimos los dos tipos de criptografías actuales dependen de la seguridad de sus llaves.

2.1 La administración de llaves

Schneier en [3] establece que una de las partes más difíciles de la criptografía es la administración de llaves, de igual forma Kaeo en [4] reconoce que el manejo de las llaves es un problema crítico dentro de la seguridad de los sistemas de comunicación ya que se depende de factores sociales más que técnicos. El Instituto Nacional de Estándares y Tecnología (NIST) del Gobierno de los Estados Unidos desarrolló en el

FIPS-171 (con base en el estándar X9.17 de generación de llaves) un manual de procedimiento para la administración de llaves simétricas. Actualmente se está estudiando el estándar para el manejo de llaves asimétricas.

La administración de llaves involucra su generación, autenticación, almacenamiento y distribución. La generación se refiere a la creación de las llaves de acuerdo con ciertos requerimientos; la autenticación al proceso de validar el origen que hace la petición del servicio de las llaves; el almacenamiento involucra salvaguardar la integridad, disponibilidad y confidencialidad de las llaves durante su tiempo de vida y la distribución es el proceso de transferir las llaves de manera segura a las entidades participantes.

La generación de llaves es un proceso crítico ya que de nada nos sirve tener un algoritmo robusto si se tiene una llave débil. Es importante señalar que toda información cifrada es sujeta a un ataque de fuerza bruta, es decir, probar todo el espacio de llaves. Un ataque más adecuado es el de diccionario en el que solo se prueba el espacio más común. Por ejemplo, DES utiliza una llave de 56 bits, lo cual nos da un espacio de llaves de 2^{56}. Sin embargo, un método pobre de generación de llaves puede reducir el espacio hasta 2^{40} lo cual implica hacer un ataque de fuerza bruta 10 mil veces más sencillo.

Cuando la generación de llaves se delega a los usuarios, éstos por lo general escogen llaves débiles lo cual hace más fácil el comprometer la seguridad de la información. Es importante señalar que el algoritmo sigue siendo el mismo y la fortaleza del sistema depende directamente de la llave. Podemos tener un algoritmo muy robusto pero si elegimos una llave débil todo el sistema será débil. Recordemos que nuestra seguridad es tan fuerte como el más débil de nuestros eslabones, y definitivamente en la generación de las llaves se tiene un eslabón bastante débil. Sin embargo, debemos decir que la solución es sencilla, es más ni siquiera se necesita un sofisticado método para la elección de la llave, sólo se requiere que ésta tenga dimensión y entropía acorde a la sensibilidad de nuestra información.

Una solución más compleja al problema de la generación de llaves es utilizar un generador de números pseudo aleatorios. Sugerimos Yarrow [5] desarrollado por Bruce Schneier, en el caso de Linux sugerimos utilizar /dev/random el cual es un dispositivo que genera números pseudo aleatorios, se incorporó a partir del kernel 2.0.31. Como comentaba Kaeo en [4], efectivamente se involucra el factor humano dentro de la elección de las llaves, por lo que se tiene que ser severo en la política para asegurar que éste no será nuestro eslabón más débil.

La generación de llaves es un factor que puede comprometer la seguridad de nuestro esquema, ya que puede ser que la información cifrada resida en el sistema conectado al ambiente hostil, entonces podría ser robada, recordar que partimos del hecho que nuestro sistema puede ser penetrado, y aplicársele un ataque de fuerza bruta o de diccionario. La solución a este problema es que la información cifrada también puede ser distribuida a través de secretos compartidos (se explica más adelante) por lo que ya no reside físicamente en el sistema.

El problema de la distribución y autenticación de las llaves ha sido solucionado por la criptografía asimétrica, toda llave pública (utilizada para el cifrado) tiene una y solo una pareja llamada llave privada (utilizada para el descifrado). Esta última a su vez es protegida por una llave simétrica.

El almacenamiento de las llaves simétricas casi siempre se lleva a cabo fuera del sistema. En el mejor de los casos, los usuarios las memorizan y en el peor las apuntan en un papel. Sin embargo, en caso de que se olviden dichas llaves entonces la información será inaccesible. Por esto se requiere un respaldo seguro de las llaves, más adelante cuando describamos nuestra propuesta profundizaremos en este aspecto.

En el caso del almacenamiento de las llaves privadas, éstas por lo general residen en el servidor conectado al ambiente hostil y solo están protegidas con otra llave que es simétrica. Debido a que gran cantidad de información se está cifrando con una sola llave pública y por lo tanto solo puede ser descifrada con su respectiva llave privada, es necesario establecer un esquema de almacenamiento seguro de dicha llave ya que se comprometa dicha llave implica que se comprometa una gran cantidad de información. Esta es la principal motivación de nuestro esquema además de que los actuales protocolos de comunicación segura a través de Internet utilizan precisamente criptografía de llave asimétrica, como lo son SSL, SSH, PGP.

3. Secretos Compartidos

Adi Shamir en [1] propone una técnica para dividir un secreto en N partes, de tal manera que el tener una de estas partes no da ninguna información. Son necesarias M de N partes para poder recuperar el secreto. Se establece que esta técnica puede ser utilizada para brindar seguridad a las llaves, nosotros pensamos que tiene mayores alcances ya que puede servir además para darle seguridad, bajo ciertos parámetros, a cualquier tipo de información. Las mejores características de dicha teoría son que no depende de una contraseña y que necesita M de N partes para recuperar la información, donde M<N (figura 1).

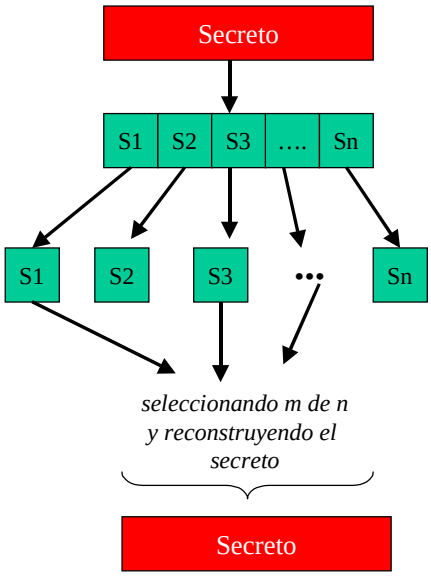

Figura 1. Esquema general de secretos compartidos

El almacenar la información en un solo lugar crea el problema del llamado punto único de falla, además si el sistema es penetrado se tiene acceso a toda la información. El posible mal funcionamiento del sistema compromete la integridad de la información. Es por esto que proponemos la técnica de secretos compartidos para dividir la información sensitiva, en nuestro caso las llaves, en N partes. Cada una de estas N partes será enviada a N sistemas (N1, N2, N3, …, Nn) dentro de una red privada. Al implementar esta técnica podemos garantizar la integridad de la información ya que no estará físicamente en el sistema original (expuesto a una red pública) sino distribuida en N sistemas dentro de una red privada. También aumentamos su disponibilidad ya que pueden corromperse hasta N-M partes y todavía puede recuperarse la información y finalmente obtenemos bajo un ambiente distribuido confidencialidad de la información, ya que el tener M-1 partes no me permite tener acceso a la información.

La teoría de los secretos compartidos cuenta con las siguientes propiedades:

- Para recuperar la información original es necesario tener un número significativo de las partes.
- La información original debe poder recuperarse aún cuando una porción significativa de las partes haya sido comprometida.
- El que recibe una de las partes debe poder verificar que la división de la información se ha realizado de manera correcta.

El esquema original propuesto por Adi Shamir sólo satisface las dos primeras propiedades, razón por la cual se introdujo un nuevo concepto basado en la misma teoría, llamado Secretos Compartidos Verificables. Con ésta se puede satisfacer la tercera propiedad. Cada una de las partes puede verificar que

tiene una parte de la información que permitirá el reconstruir la información original, aunque la información que tiene no le permite conocer o intuir la información original. Esta teoría ha sido ampliamente estudiada por Rosario Gennaro en [2].

Uno de los problemas de la teoría de los secretos compartidos es que la información original debe ser recuperada en un cierto lugar [3], si dicho lugar es el sistema que está expuesto a la intrusión obviamente la integridad de la información se vería comprometida. Sin embargo, para nuestra aplicación no es necesario que la información original se recupere en dicho sistema, de hecho puede ser recuperada en cualquiera de los sistemas que cuenta con uno de los secretos compartidos (N1, N2, N3, …, Nn). Aún más, podemos respaldar cada uno de los secretos compartidos y procesarlos en un sistema físicamente desconectado de todo nuestro esquema.

4 Nuestra Propuesta

Nuestra propuesta consiste en utilizar la teoría de secretos compartidos para distribuir las llaves criptográficas entre N sistemas que residen en una red que consideramos privada (no está conectada físicamente al ambiente hostil). Mediante esta propuesta podemos garantizar la integridad de la llaves criptográficas ya que el intruso tendría que penetrar M sistemas para poder tener acceso a dicha información. En un inicio proponemos el esquema (M,N) donde M igual con 2 y N igual con 3, es decir, las llaves criptográficas se dividen en 3 partes y se requerien 2 partes para reconstruirlas. Desde otro punto de vista, el intruso requiere penetrar primero un sistema, conectado al ambiente hostil, y para tener acceso a las llaves de dicho sistema requiere penetrar otros dos dentro del ambiente privado. Es importante recordar que una característica esencial de los secretos compartidos es que una sola parte no proporciona información sobre la original, se requieren por lo menos M partes. Y, desde otro punto de vista, puede fallar uno de los tres sistemas y aún así podemos recuperar de manera íntegra las llaves, con esta propiedad aumentamos la disponibilidad de nuestra información.

Las principales ventajas de nuestra esquema son que se puede garantizar la integridad de la llaves criptográficas ya que éstas ya no están dentro del sistema sino en N sistemas dentro de una red privada por lo que el intruso no podrá borrarlas, ni modificarlas, ni siquiera observarlas. El hecho de no estar físicamente presentes en el sistema conectado al ambiente hostil les proporciona confidencialidad. Por último aumentamos la disponibilidad de las llaves ya que puede fallar un sistema y aún así se puede acceder a la información original.

Dicho esquema es muy flexible ya que podemos aumentar tanto M como N de acuerdo a nuestros requerimientos. En un ambiente muy hostil podemos proponer M igual con 3 y N igual con 5, con lo cual necesitamos 3 partes de 5 para recuperar nuestras llaves. Este esquema también se puede implementar en un ambiente que requiera alta disponibilidad, para lo cual se requiere que M sea relativamente pequeño y N relativamente grande. En [13] se encuentra un estudio sobre el tamaño de M y N.

Con el esquema M igual a 2 y N igual con 3, si un intruso penetrara el sistema A, no encontrará las llaves. Una vez estando en el sistema A intentará buscar en otros sistemas las llaves, bajo nuestro esquema requerirá penetrar por lo menos otros dos sistemas y además aplicar el algoritmo para recuperar la información original. Además tendrá que violar los controles de acceso que se apliquen en el sistema A para poder acceder a la red segura. Dichos controles comprenden el filtrado de tráfico de A solo hacia ciertos sistemas y el tráfico de los sistemas de la red segura hacia A. Para complicar el esquema el puerto por el que se comunica A con los sistemas dentro de la red segura cambian con el tiempo. Como se comentó en un principio podemos aumentar el número de M y N en caso de que se considere un ambiente más hostil, lo cual complicaría en gran medida el acceso a las llaves ya que ahora se tendrán que penetrar M sistemas para poder tener acceso a la información original.

En primera instancia hemos elegido las llaves criptográficas como elemento a proteger, sin embargo, este esquema también puede servir para salvaguardar la integridad y la confidencialidad de la información sensitiva dentro del sistema, por ejemplo, el archivo de contraseñas, el respaldo del contenido público de nuestro servidor de web (utilizado en esquemas de restablecimiento automático de información en caso de

ser modificada, ej. Tripwire), información de los usuarios del sistema, huellas de los distintos programas utilizados (con lo cual podemos evitar caballos de troya y códigos maliciosos), bases de datos, llaves privadas y secretas, así como documentos con información clasificada.

Recomendamos utilizar una tercera entidad en la que se reconstruya la información original, es decir, la llave. Ya que si se reconstruye en el sistema que está expuesto al ambiente hostil corremos el riesgo de que dicha llave sea comprometida. Esta tercera entidad actuará como interfaz y administrador entre el ambiente hostil y nuestros N sistemas dentro de la red segura. Reconocemos que esto implica el introducir un punto único de falla, sin embargo, la integridad y confidencialidad de la llave se garantiza. Esta tercera entidad donde se recupera la llave también puede ser uno de los N sistemas.

5 Implementación

Lo primero a definir es la información a almacenar dentro de las llaves criptográficas generadas en el sistema A. En [4] se establecen los principales puntos que deben auditarse en un sistema basado en un extensa investigación para obtener información suficiente que nos lleve a conclusiones adecuadas sobre la intrusión a un sistema.

Entre las principales recomendaciones se establece que las acciones deben ser auditadas al principio de su ejecución y no al final como se acostumbra hacerse. Así mismo es necesario establecer los parámetros introducidos en ciertos programas para conocer las intenciones reales de un intruso, por ejemplo, el programa de búsqueda no manda a la bitácora lo que se está introduce en el campo de la búsqueda. Los mecanismos que auditan el uso y ejecución del comando *"su"* no están bien establecidos ya que pueden llegar a enmascararse con lo que un intruso podría no llegar a detectarse.

Además de estas consideraciones, se recomienda habilitar las bitácoras necesarias para que un sistema sea considerado con nivel de seguridad C2, avalado por el libro naranja (TCSEC, Trusted Computer System Evaluation Criteria). En este se establece la necesidad de auditar: logins, logouts, accesos remotos al sistema, apertura, cerrado, renombrado y borrado de archivos, cambios en los privilegios y atributos de seguridad de los archivos y programas. Además se requiere capturar para cada uno de los eventos anteriores, la fecha y hora del evento, la identificación única del usuario que inició el evento, el tipo de evento, éxito o fallo del evento, el origen de la petición, el nombre del objeto involucrado, descripción de las modificaciones a las base de datos de seguridad.

Nuestro esquema requiere el establecer dos tipos de redes, la primera una red hostil o pública y la segunda una red segura o privada a la que se le pueden aplicar controles de acceso y un monitoreo muy cercano. El sistema A, conectado al ambiente hostil, fue implementado en una máquina que consta de dos tarjetas de red, una conectada al ambiente hostil y otra conectada a la red segura. Los sistemas de la red segura R1, R2, ..., Rn (que contienen los secretos compartidos) se implementaron en tres máquinas de la misma red. Las cuatro máquinas forman parte de una red local, formada por varias computadoras IBM Intel stations, pentium III a 500 mhz y con 128M en RAM. Cada computadora cuenta con el sistema operativo *"Security-Enhanced Linux*" desarrollado por la Agencia de Seguridad Nacional de los Estados Unidos. Si bien es cierto que las computadoras cuentan con una buena capacidad de computo, los sistemas R1, R2, ..., Rn no requieren gran poder de procesamiento ya que solo escuchan por un puerto para recibir la información enviada por A. El espacio en disco no es un factor muy importante ya que el tamaño de las llaves no es muy grande. Lo anterior se encuentra ilustrado en la figura 2.

El sistema A contiene el programa que toma las llaves, les aplica el algoritmo de secretos compartidos y las N partes las envía a través de sockets por un puerto que cambia conforme al tiempo a R1, R2, ... Rn. Estos últimos cuentan con un programa recibe y almacena la información enviada por A. La información de las llaves es recuperada, para su análisis ya sea forense o de detección de intrusos, en un equipo ajeno a todo el sistema. Una opción alterna es que la información se almacene en uno de los R1, R2, ..., Rn sistemas. No se recomienda hacer dicha operación en el sistema A, puede haber un intruso en el sistema, que podría comprometer el sistema.

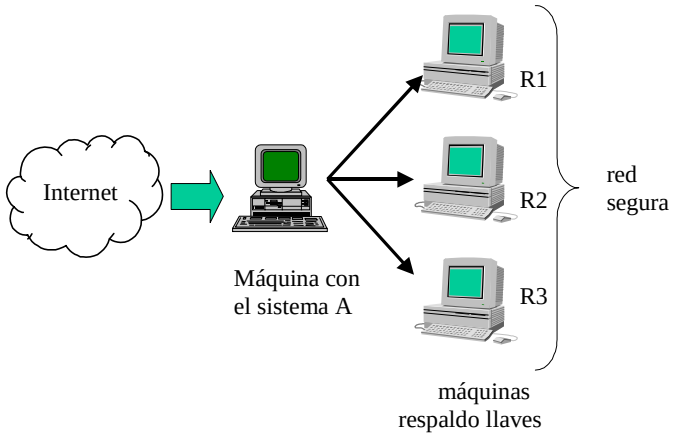

Figura 2. Esquema general del sistema

La forma en que se genera la llaves va mas alla de los propósitos de este trabajo. Se considera que las llaves se encuentran en el sistema A . Todos los sistemas (A, R1, R2, …, Rn) se les realizó una auditoría inicial de todos sus archivos y programas y se obtuvo una huella digital (MD5) de los mismos (a través del software Tripwire). Dichas huellas se guardaron en un medio de solo lectura para garantizar su integridad y poder cotejar en caso de un incidente. Además se realizaron monitoreos con herramientas como Nessus, Cops y Saint para detectar y corregir las vulnerabilidades más comunes. Se ha instaló Tcp Wrappers en los sistemas R1, R2 y R3 para que solo puedan ser accedidos bajo ciertas direcciones, protocolos y puertos. Se instaló Xinetd el cual es una extensión del demonio de internet presente en la mayoría de los sistemas Linux con la ventaja de que éste genera bitácoras.

El desempeño de las máquinas no se vio afectado por el sistema y la reconstrucción de las llaves no tomo mucho tiempo (unos 2 segundos).

6. Conclusiones y trabajo futuro

Se presentó un esquema que permite garantizar la integridad de las llaves de un sistema. El esquema se basa en la teoría de secretos compartidos propuesta por Adi Shamir y fue implementado en un ambiente distribuido.

Nuestro esquema permite una gran flexibilidad que permite adaptarse a distintos escenarios dependiendo el grado de hostilidad al cual nos estemos enfrentando, así como al grado de disponibilidad que requerimos. Este esquema no represento ninguna carga considerable al sistema.

Aun restan varias pruebas por hacer, tan sólo se hicieron pruebas a nivel reconstrucción con dos de tres máquinas y con información “ligera”. Lo próximo a realizar es probar el sistema con un conjunto más grande de computadoras, así como en un equipo de producción. Este tipo de pruebas nos llevará a definir parámetros que permitan establecer el valor correcto de los parámetros M y N de la teoría de secretos compartidos. También se esta explorando el utilizar el esquema de secretos compartidos con otro tipo de información crítica del sistema.

Referencias

[1] A. Shamir, “How to share a secret”, Comm. of the ACM, Vol. 22, 1979, pp 612-613.
[2] S. Garfinkel, G. Spafford. “Practical Unix and Internet Security”. O’Reilly & Associates Inc. 1996, ISBN 1-56592-148-8
[3] B. Schneier. “Applied Cryptography” . John Wiley and Son, Inc. 1996, ISBN 0-471-12845-7
[4] M. Kaeo. “Designing Network Security”. Cisco Press, 1999. ISBN 1-57870-043-4

[5] J. Kelsey, B. Schneier, and N. Ferguson. “Yarrow-160: Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator”.
[6] S. Axelsson, “An approach to Unix Security Logging”,
[7] R. Gennaro, “Theory and Practice of Verifiable Secret Sharing”,
[8] B. Schneier, “Cryptographic Support for Secure Logs on Untrusted Machines”, The Seventh USENIX Security Simposium Proceedings, USENIX Press, Jan 1998, pp. 53-62.
[9] W. Stallings. “Cryptography and Network Security”. Prentice Hall Inc. ISBN 0-13-869017-0
[10] H. Tipton, M. Kruse. “Information Security Management Handbook”. Auerbach Publications. ISBN 0-8493-9829-0
[11] D. Russell, G.T. Gangemi. “Computer Security Basics”. O’Reilly & Associates Inc. ISBN 0-937175-714
[12] N. Koblitz. “A Course in Number Theory and Cryptography”. Springer . ISBN 3-540-94293-9
[13] A. de Santis, L. Gargano, U. Vaccaro. “On the size of shares for Secret Sharing Schemes”, J. Cryptology 6 (1993), 157-167. [Preliminary version appeared in "Advances in Cryptology --CRYPTO '91", J. Feigenbaum, ed., Lecture Notes in Computer Science 576 (1992), 101-113.]

Secure Storage Scheme for Cryptographic Keys

Roberto Gómez Cárdenas, Ricardo C. Lira Plaza, Adolfo Grego

Computer Science Department
Instituto Tecnológico y de Estudios Superiores de Monterrey – Campus Edo. de México
Apdo. Postal 6-3, Módulo Servicio Postal, Atizapán C.P. 52926, Edo. México
{rogomez, rlira, agrego}@campus.cem.itesm.mx

Abstract: Computer security is defined as the mechanisms and policies set that assures confidentiality, integrity and availability of systems resources. In our days many cryptosystems are used in order to secure systems, however they depends on a cryptographic key. Our work proposes a scheme, based in secret sharing theory that assures the integrity and confidentiality of cryptographic keys. Furthermore our proposition allows fault tolerance, so the availability feature of the data lost is also considered.

Keywords: cryptology, shared secrets, computer security, networks, distributed systems