Seminario – Seguridad en desarrollo del Software

Tema: Criptografía I

Autor: Leudis Sanjuan

¿Qué es criptografía?

La criptografía es el nombre genérico con el que se designan dos disciplinas opuestas y a la vez complementarias: criptografía y criptoanálisis. La criptografía se ocupa del diseño de procedimientos para cifrar; es decir, para enmascarar una determinada información de carácter confidencial. El criptoanálisis se ocupa de romper esos procedimientos para así recuperar la información. Ambas disciplinas siempre se han desarrollado de forma paralela, pues cualquier método de cifrado lleva siempre emparejado su criptoanálisis correspondiente.

El esquema fundamental de un proceso criptográfico (cifrado/descifrado) puede resumirse como muestra la siguiente figura:

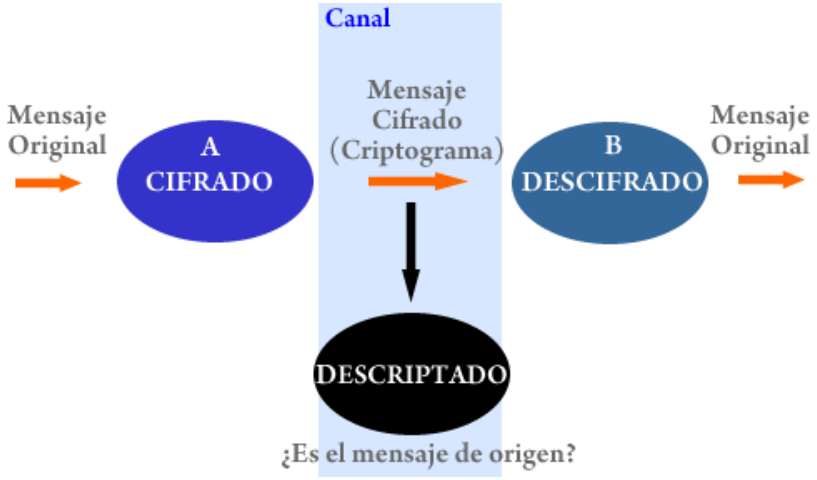

Figura1: Proceso criptográfico

A y B son respectivamente el emisor y el receptor de un determinado mensaje. A transforma el mensaje original mediante un determinado procedimiento de cifrado controlado por una clave, en un mensaje cifrado (llamado también criptograma) que se envía por un canal público. En recepción, B con conocimiento de la clave transforma ese criptograma en el mensaje original.

En el proceso de transmisión, el criptograma puede ser interceptado por un enemigo criptoanalista que lleva a cabo una labor de desencriptado; es decir,

intenta, a partir del criptograma y con el conocimiento de la clave, recuperar el mensaje original. Un buen sistema criptográfico será, por tanto, aquel que ofrezca un descifrado sencillo, pero un desencriptado imposible o muy difícil.

En sus principios, la criptografía (llamada hoy en día criptografía clásica) se ocupaba de mantener la confidencialidad del mensaje; hoy en día, la criptografía se enfoca en el concepto de comunicaciones seguras y a la vez busca cumplir con tres propósitos:

- Mantener la confidencialidad del mensaje; es decir, que solo sea visto por aquellos que tienen que ver la información
- Garantizar la autenticidad del destinatario y remitente; es decir, asegurar la identidad del destinatario y remitente.
- Garantizar la integridad en el mensaje; es decir, que el mensaje enviado por el emisor sea el mismo que reciba el receptor.

Tipos de sistemas criptográficos

Teniendo en cuenta el tipo de clave, podemos distinguir dos métodos criptográficos:

1. **Sistemas de clave única o métodos simétricos**: son aquellos en los que los procesos de cifrado y descifrado son llevados a cabo por una única clave.

2. **Sistemas de clave pública o asimétrica**: son aquellos en los que los procesos de cifrado y descifrado son llevados a cabo por dos claves distintas y complementarias.

A continuación explicaremos cada uno de estos sistemas

Sistemas de clave única o métodos simétricos

Esta forma de cifrado utiliza una clave secreta, denominada secreto compartido, compartida por emisor y receptor. El receptor necesita la clave secreta para desbloquear los datos, esto lo hace por medio de un algoritmo de cifrado. Se denomina criptografía simétrica porque tanto para cifrar como para descifrar se necesita la misma clave.

Dentro de los sistemas simétricos distinguimos dos tipos de algoritmos:

- **Cifrado por bloque:**

  Es aquel en el que se cifra el mensaje original agrupándolo en bloques de tamaño fijo, por ejemplo 64 bits.

- **Cifrado por flujo:**

  Es aquel en el que se cifra el mensaje original bit a bit o byte a byte.

Por otro lado, los sistemas de cifrado simétrico presentas dos grandes desventajas: la distribución de las claves (en un medio público, el cual puede ser interceptado) y la dificultad de almacenar y proteger muchas claves diferentes.

Sistemas de clave pública o asimétrica

Esta forma de cifrado utiliza dos claves: una clave es secreta y una clave pública. El mensaje lo ciframos con la clave pública del destinatario. Este puede descifrar a continuación con su propia clave privada. La diferencia de este sistema es que nadie necesita la clave privada de otro para poder enviar un mensaje en forma segura. Utilizamos su clave pública, la cual no necesita mantenerse segura. Al utilizar la clave pública del destinatario, sabemos que sólo esa persona puede cifrar utilizando su propia clave privada.

Este sistema tiene algunas desventajas: para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso, las claves deben ser de mayor tamaño que las simétricas y el mensaje cifrado ocupa más espacio que el original.

Por otro lado, teniendo en cuenta el tipo de operación que es usado para transformar el mensaje original en un mensaje cifrado, podemos distinguir dos métodos criptográficos:

- **Cifrado por sustitución**

Este método consiste en establecer una correspondencia entre las letras del alfabeto en el que está escrito el mensaje original y los elementos de otro conjunto, que puede ser el mismo o distinto alfabeto. De esta forma, cada letra del texto original se sustituye por un símbolo correspondiente en la elaboración del criptograma. El receptor por su parte, conoce la correspondencia establecida, y sustituye cada símbolo del criptograma por el símbolo correspondiente del alfabeto original, recuperando así el mensaje emitido originalmente. Ejemplo:

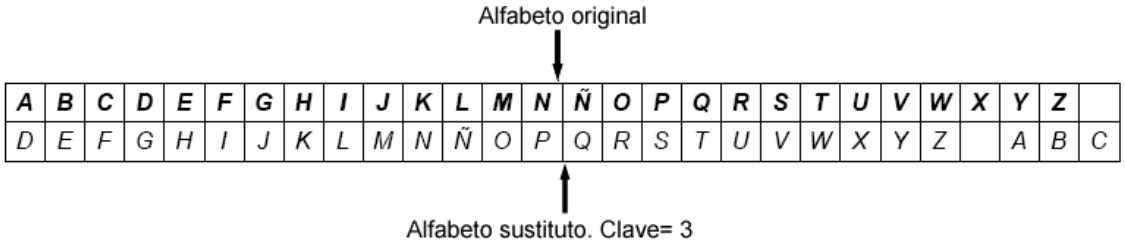

Figura 2: Ejemplo básico de criptografía por sustitución

Con este tipo de criptografía, la frase: CASA

Pasaría a ser: FDVD

- **Cifrado por transposición**

Consiste en reorganizar los símbolos del mensaje original colocándolos en un orden diferente, de tal forma que el criptograma contengan los mismos elementos del mensaje original, pero colocándolos de tal forma que resulten incomprensibles. El receptor, con conocimiento de la transposición, organiza los símbolos desordenados del criptograma en su posición original. Por ejemplo:

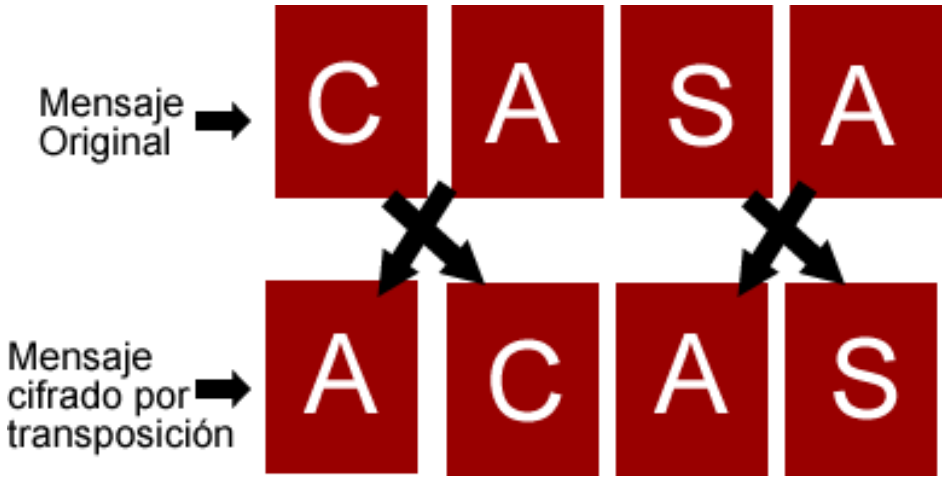

Figura 3: Ejemplo básico de criptografía por transposición

Por último, es importante señalar que el cifrado por sustitución y transposición eran los métodos usados por la criptografía clásica.

Algoritmos de cifrado

Los algoritmos de cifrados son programas que realizan el proceso de criptografía basándose en los tipos de cifrado. A continuación mencionaremos los algoritmos de cifrado más usados para el proceso de encriptación.

Algoritmos criptográficos simétricos

- **DES (*Data Encryption Standard*):** Es un algoritmo de cifrado por bloques de 64 bits. Fue ideado por IBM y aceptado por el NIST (*National Institute of Standars and Technology*) en 1976. Se trata de un algoritmo de 64 bits de clave de los cuales 56 bits componen la clave de cifrado propiamente dicha, mientras los 8 restantes son de paridad y se usan para corrección de errores.

  DES tiene como entradas, para su funcionamiento, un dato y una clave de 64 bits cada uno. Al inicio y final del algoritmo se aplican dos permutaciones al dato que poseen la característica particular, que una es la inversa de la otra. Luego de la aplicación de la primera permutación el dato es pasado por dieciséis rondas de cifrado en las cuales se hace necesaria la utilización de unas subclaves que son obtenidas a partir de la clave que se ingresa al inicio del algoritmo, en total son dieciséis subclaves, una para cada ronda. Al terminar el total de las rondas de cifrado se aplica la ultima permutación y de esta manera se obtiene un dato cifrado por medio del algoritmo DES.

  DES actualmente ya no es estándar criptográfico y fue roto en enero de 1999 con un sistema de cómputo que analizaba 250.000.000.000 claves por segundo.

  Su principal ventaja es la rapidez de cálculo y la sencillez de su implementación. Sus principales desventajas son la poca longitud de clave que maneja y la incapacidad de manejar claves de longitud variable.

  Para mayor información, visitar: http://www.itl.nist.gov/fipspubs/fip46-2.htm

- **Triple-DES (*Triple - Data Encryption Standard*):**dada la capacidad de cómputo actual y la relativa facilidad que supone romper el algoritmo DES, se desarrolló DES TRIPLE, el cual consiste en aplicar tres veces el algoritmo DES en un orden específico. Primero se cifra el dato con una clave, el resultado de esto es descifrado con otra clave y por último el resultado del descifrado es cifrado nuevamente. La clave que se emplea en este último paso puede ser la primera clave utilizada o puede ser una nueva clave.

  Mediante este sistema se obtiene un cifrado de 192 bits (168 efectivos y 24 de paridad) con tres claves que resulta mucho más complejo de vulnerar.

  Para mayor información, visitar: http://en.wikipedia.org/wiki/Triple_DES

- **AES (*Advanced Encryption Algorithm*):** también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado para el gobierno de los Estados Unidos. Actualmente es uno de los algoritmos más populares usados en criptografía simétrica.

  Este algoritmo cifra bloques de longitudes 128, 192 o 256 bits. Con la característica que definido el tamaño del bloque, usa todas las claves de la misma longitud. Toda la información para este algoritmo es basada en bytes, cifra mensajes de longitudes 16, 24 o 32 bytes con claves de longitud 16, 24 o 32 bytes. Los mensajes y claves se manejan en forma de matrices con 4 filas; en donde, los mensajes son matrices de 4 × Nb bytes, siendo Nb = 4, 6, 8. Por su parte la clave es una matriz de 4 × Nk, siendo Nk = 4, 6, 8.

  Para mayor información, visitar: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

- **IDEA (*International Data Encryption Algorithm*):** fue creado por Xuejia Lai y James Massey en 1990. Es un cifrado de bloque, que opera sobre mensajes de 64 bits con una clave de 128 bits. Consiste en ocho transformaciones idénticas (cada una llamada una ronda) y una transformación de salida (llamada media ronda). Gran parte de la seguridad de IDEA deriva del intercalado de tres operaciones: Operación O-exclusiva (XOR) bit a bit, suma módulo 216 y multiplicación módulo 216+1.

  Para mayor información, visitar: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Algoritmos criptográficos asimétricos

[1]**RSA (Rivest - Shamir - Adleman):** en criptografía, RSA es un sistema criptográfico de clave pública desarrollado en 1977 por Rivest, Shamir y Adelman (de aquí el nombre RSA).

La seguridad de este algoritmo radica en el problema de la dificultad para factorizar grandes números enteros. Los mensajes enviados se representan mediante números y el funcionamiento se basa en el producto (conocido) de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de 10200, y se prevé que su tamaño aumente con el aumento de la capacidad de cálculo de los ordenadores.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada.

Para mayor información, visitar: http://en.wikipedia.org/wiki/RSA

Diffie-Hellman: La seguridad de Diffie-Hellman se basa en la función con trampa del problema del logaritmo discreto. Se emplea generalmente como medio para acordar claves simétricas que serán empleadas para el cifrado de una sesión.

El funcionamiento de este algoritmo se resume a continuación:

[1] Tomado de : http://es.wikipedia.org/wiki/RSA

- Sea **q** un número primo muy grande
- Sea $\propto$ una primitiva de **q**
- Ana elige un número A y transmite XA= $\propto$ A mod q
- Juan escoge un número B y transmite XB = $\propto$ B mod q
- Ana calcula la clave de sesión K = (XB)A modulo q
- Juan calcula la clave de sesión K = (XA)B modulo q
- Se establece sesión con la clave K

Para mayor información, visitar: http://es.wikipedia.org/wiki/Diffie-Hellman

Algoritmos Hash

Los algoritmos Hash, o de resumen, se constituyen un tipo especial de criptosistemas. Esto,s a diferencia de los algoritmos simétricos o asimétricos, no utilizan el concepto de clave. Para estos algoritmos existe un nuevo término llamado: *fingerprint* o huella digital o resumen o *hash*

Una función Hash toma un mensaje de entrada de longitud arbitraria y genera un código de longitud fija. La salida de longitud fija se denomina *hash* del mensaje original.

Los criptosistemas Hash presentan las siguientes características:

- Unidireccionalidad: este concepto significa que deberá ser computacionalmente muy difícil, por no decir imposible, obtener el mensaje M (original).

- Compresión: a partir de un mensaje de cualquier longitud, el *hash* H(M) debe tener una longitud fija. Normalmente mucho menor.

- Coherente: la misma entrada (mensaje original) siempre deberá producir la misma salida (mensaje original).

- Facilidad de Cálculo: debe ser fácil calcular la función Hash H(M) a partir de un mensaje M.

- Único: casi imposible encontrar dos mensajes que generen el mismo *hash.*

- Difusión: el resumen H(M) debe ser una función compleja de todos los bits del mensaje M.

Para mejor compresión, a continuación ilustraremos el funcionamiento de estos tipos de criptosistemas:

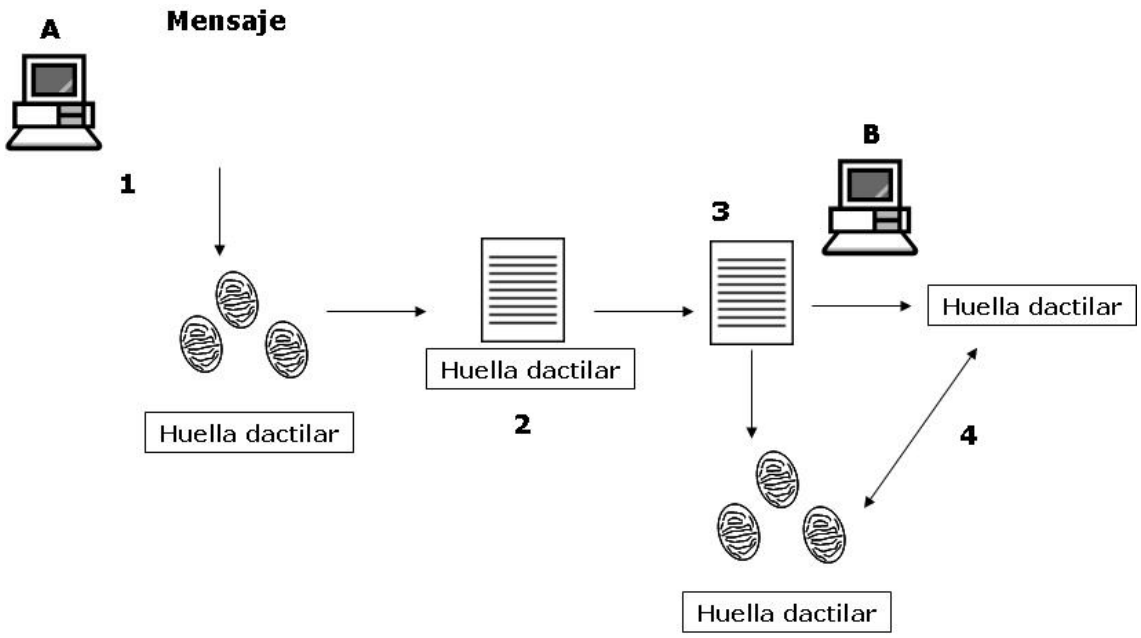

Figura 4: Algoritmos Hash

1. A escribe un mensaje y lo utiliza como la entrada de una función *hash.*
2. El resultado de la función *hash* se añade como huella dactilar al mensaje que se envía a B.
3. B separa el mensaje y la huella dactilar adjunta y utiliza el mensaje como entrada de la misma función *hash* que utilizó A.
4. Si las *hash* coinciden , B puede estar seguro que el mensaje no ha sido modificado.

A continuación vemos algunos de los principales algoritmos criptográficos de tipo *hash*:

[2]MD5 (M*essage Digest Algorithm 5*, Algoritmo de Ordenación de Mensajes 5): es un algoritmo desarrollado por RSA *Data Security, Inc*. MD5 es una función *hash* de 128 bits, que toma como entrada un mensaje de cualquier tamaño y produce como salida un resumen del mensaje de 128 bits.

El primer paso del algoritmo divide el mensaje en bloques de 512 bits. El último bloque o si el mensaje completo es menor a 512 bits, se formatea para tener un tamaño de 512 bits mediante el agregado de bits 0 más la longitud del tamaño del mensaje.

Además, se tiene un búfer estado de 128 bits manejado como cuatro palabras de 32 bits. La función compresión tiene cuatro rondas y en cada ronda el bloque de mensaje y el búfer son combinados en el cálculo, mediante el uso de sumas modulares, XOR's, AND's, OR's y operaciones de rotaciones sobre palabras de 32 bits.

[2] Tomado de : http://ccc.inaoep.mx/~rcumplido/papers/ReCOnFig04%20-%20MD5.pdf

Cada ronda combina el bloque de 512 bits del mensaje con el búfer estado, así que cada palabra del mensaje es usado cuatro veces. Después de las cuatro rondas de la función compresión, el búfer estado y el resultado son sumados (sumas módulo 232) para obtener la salida

SHA (*Secure Hash Algorithm*): la familia SHA (*Secure Hash Algorithm*, Algoritmo de Hash Seguro) es un sistema de funciones *hash* criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el *National Institute of Standards and Technology* (NIST).
El primer miembro de la familia fue publicado en 1993 y fue llamado oficialmente como SHA. Sin embargo, hoy en día, se le llama SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (todos ellos son referidos como SHA-2).

SHA-1 ha sido examinado muy de cerca por la comunidad criptográfica, y no se ha encontrado ningún ataque efectivo. No obstante, en el año 2004, un número de ataques significativos fueron divulgados sobre funciones criptográficas de *hash* con una estructura similar a SHA-1; esto ha planteado dudas sobre la seguridad a largo plazo de SHA-1.
SHA-0 y SHA-1 producen una salida resumen de 160 bits de un mensaje, que puede tener un tamaño máximo de 264 bits, y se basa en principios similares a los usados MD5.

SHA-2 produce una salida resumen de 256 (para SHA-256) o 512 (para SHA-512) y difiere a SHA-1 en que el algoritmo contempla algunas constante adicionales; así mismo, el tamaño del resumen es diferente al igual que el número de rondas.

Ataques contra los sistemas criptográficos

Así como existen algoritmos que protegen la confidencialidad y la integridad de la información, existen técnicas o métodos que tratan de romper la seguridad que proporcionan este tipo de algoritmos.

Un ataque contra los sistemas criptográficos puede tener alguno de los siguientes objetivos:

- Descubrir el mensaje original.
- Alterar el mensaje M por M' y lograr que el receptor acepte M' como mensaje original del emisor.
- Iniciar una comunicación con un receptor y que el atacante sea catalogado como un emisor autorizado. Esto se llama "*spoofing*".

Se define como criptoanálisis a la ciencia que enmarca un conjunto de técnicas cuyo objetivo consiste en descifrar mensajes cifrados sin conocer las claves correctas. A continuación se explican cada una de estas técnicas:

Ataques de búsqueda de llave (fuerza bruta): la forma más fácil de romper un código es probar todas las llaves posibles. Estos ataques consisten en probar una por una de las posibilidades hasta encontrar la llave correcta. No hay forma de protegerse contra estos ataques, ya que no hay forma de evitar que un atacante intente desencriptar un mensaje probando las n opciones. Los ataques de búsqueda de llaves no son efectivos, algunas veces ni siquiera son posibles ya que hay demasiada llaves que probar y no hay tiempo para probarlas todas.

A nivel de procesamiento computacional son muy pesados ya que hay que probar todas las opciones.

Ataques sólo a Texto Cifrado: el atacante no conoce nada acerca de los contenidos del mensaje y debe trabajar a partir sólo del mensaje cifrado. En la práctica, a menudo es posible adivinar algo del mensaje sin cifrar ya que muchos tipos de mensaje tienen cabeceras de formato fijo. Incluso los documentos comunes o los correos, empiezan de una forma muy predecible. También puede ser posible adivinar que algún bloque de texto cifrado contiene una palabra común.

Ataque a Texto Sin Cifrar Conocido: el atacante conoce o puede adivinar o inferir el mensaje sin cifrar para algunas partes del texto cifrado. La tarea consiste en descifrar el resto de los bloques de texto cifrado utilizando esta información. Esto se puede hacer determinando la clave utilizada para cifrar los datos o utilizando algún "atajo".

Ataque a Texto Sin Cifrar Elegido: el atacante puede tener algo de mensaje que quiere, cifrado con la clave desconocida. La tarea consiste en determinar la clave utilizada para el cifrado. Algunos métodos de cifrado (por ejemplo, el algoritmo de clave pública o asimétrico RSA) son extremadamente vulnerables a los ataques de texto sin cifrar elegido.

Ataque *man-in-the-middle*: atacante mediante intromisión. El adversario se coloca en medio de las partes legítimas (emisor y receptor) que se comunican. Este ataque es relevante para protocolos de intercambio de claves y comunicación criptográfica. La idea es que cuando dos partes se intercambian claves para comunicaciones seguras (por ejemplo utilizando Diffie-Hellman) un adversario se coloca entre las partes en la línea de comunicaciones.

El adversario entonces realiza un intercambio de clave separado con cada parte. Las partes finalizarán utilizando una clave diferente cada una de las cuales es conocida por el adversario. El adversario entonces descifrará las comunicaciones con la clave adecuada y las cifrará con la otra clave para enviarla a la otra parte. Las partes creerán que se están comunicando de forma segura, pero de hecho el adversario está escuchando y entendiendo todo.

Criptoanálisis Diferencial, Lineal y Lineal-Diferencial: el criptoanálisis diferencial es un tipo de ataque que puede aplicarse a cifradores de bloque iterativos (como por ejemplo, los algoritmos simétricos o de clave secreta DES, IDEA, etc.).

El criptoanálisis diferencial es básicamente un ataque sobre texto sin cifrar escogido y se basa en un análisis de la evolución de las diferencias entre dos textos sin cifrar relacionados cuando se cifran bajo la misma clave. Mediante un cuidadoso análisis de los datos disponibles, las probabilidades pueden asignarse a cada una de las posibles claves y eventualmente la clave más probable se identifica como la correcta.

Ataques de diccionario: En el ataque de diccionario no se pretende obtener la clave, sino directamente el texto original.

Cuando un usuario quiere ingresar a un sistema, introduce su código y su clave de acceso, ésta se cifra y posteriormente se compara el resultado con la clave cifrada que se almacena en el archivo de claves. Si son iguales, el sistema considera que el usuario es quien dice ser y le permite el acceso.

Este tipo de programas parten del hecho de que se ha obtenido una copia del archivo de claves (por ejemplo, el denominado /etc/passwd de Unix) y comprueban si existe alguna cuenta sin clave, en cuyo caso utilizan, y con el resto se realiza el siguiente ataque.

Por una parte, se dispone de un diccionario con gran cantidad de palabras y combinaciones muy comunes y válidas como claves. Posteriormente se realiza su cifrado con la utilidad del sistema a atacar o una copia de la misma, se comprueba el resultado del cifrado con el contenido del archivo de claves y en el caso de que se produzca una coincidencia se infiere cuál es el mensaje sin cifrar.

Aplicaciones de la criptografía

Firmas digitales: Son un mecanismo que proporcionan integridad y autenticación. Las firmas digitales nos permiten saber si un mensaje lo ha enviado realmente una persona o si ha sido alterado durante el trayecto**.**

¿Cómo funcionan las firmas digitales?

A continuación se muestra un gráfico donde se resumen los pasos de generación y verificación de firmas digitales.

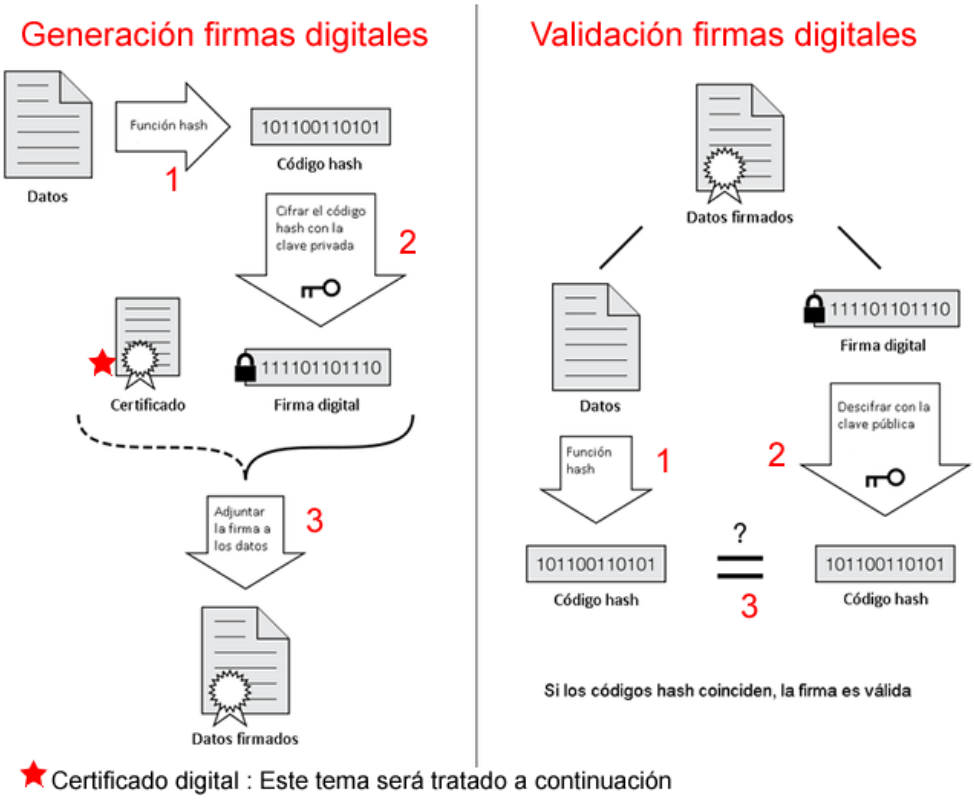

Figura 4: Generación y comprobación de firmas digitales.

Fuente: http://es.wikipedia.org/wiki/Archivo:Firma_Digital.png

Generación

1. Se aplica una función *hash* sobre el mensaje o datos iniciales.
2. El resultado de la operación anterior se le aplica un algoritmo de clave asimétrica o pública donde se utiliza la clave privada del emisor.
3. El resultado de la operación anterior genera una firma digital. A esta firma se le adjunta el certificado digital (este tema es tratado en la próxima sección) y se envían los datos firmados por la red.

Validación de firmas digitales

1. Se recibe los datos firmados, se obtienen los datos y se le aplica la función *hash* (previamente el receptor debe conocer cuál es la función *hash* que debe aplicar).
2. De los datos firmados se obtiene también la firma digital, esta firma es descifrada con la llave pública del emisor, como resultado se obtiene un código *hash*.
3. Si el código *hash* obtenido del paso 1 es igual al obtenido al paso 2, la firma es válida y por tanto hemos asegurado la autenticidad e integridad del mensaje

Certificados digitales y entidades certificadoras: en la sección anterior mencionamos que las firmas digitales utilizan la clave privada del emisor para generar la firma y la clave pública del mismo para validar la firma. Pero, ¿qué pasa si alguien envía una clave pública ajena bajo su identidad? ¿Quién asegura o da la confianza de que la clave pública de un usuario es auténtica?

La solución a este problema la trajo la aparición de los certificados digitales o certificados electrónicos. El objetivo principal de un certificado digital es garantizar con toda confianza el vínculo existente entre una persona, entidad o servidor Web con una pareja de claves correspondientes a un sistema criptográfico de clave pública.

Un certificado digital es un documento electrónico que contiene datos que identifican a una persona o entidad (empresa, servidor Web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora (AC). Las principales autoridades certificadoras actuales que existen son: VeriSign, SecureSign, GlobalSign, Thawte y CertiSign.

El formato de los certificados digitales es estándar, siendo X.509 v3 el recomendado por la Unión Internacional de Comunicaciones (ITU) y usado por la mayoría de navegadores.

X.509 V3 es un estándar para infraestructuras de claves públicas (PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación. El aspecto de los certificados X.509 v3 es el siguiente:

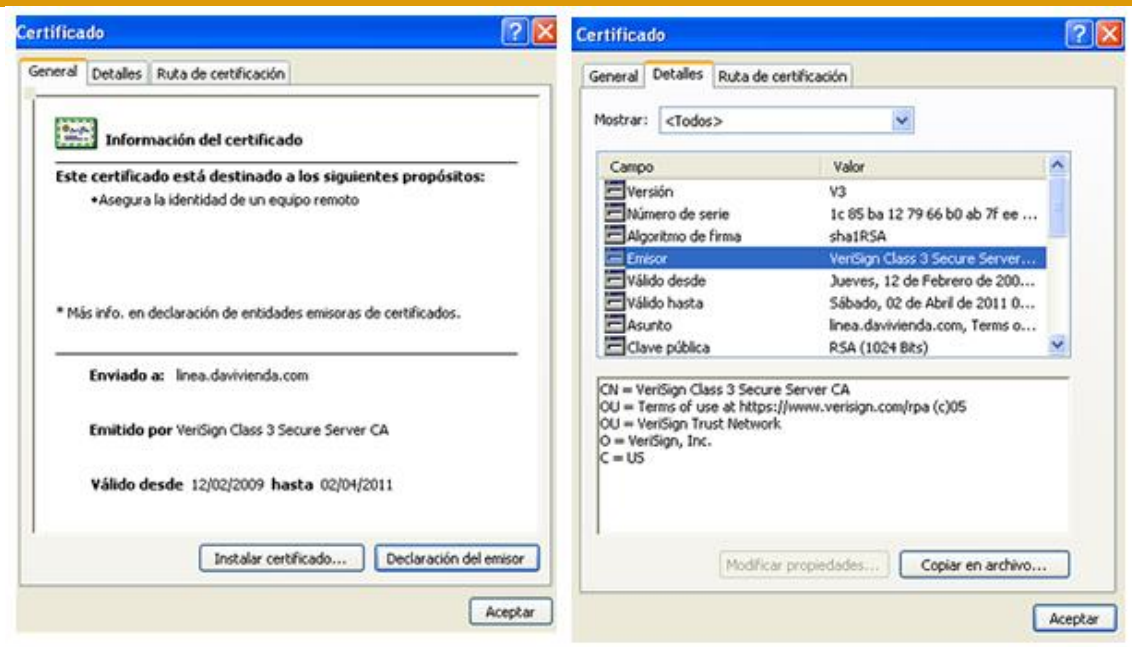

Figura 5: Aspecto certificados X.509 v3

Los datos que figuran generalmente en un certificado X.509 son:

- Versión: versión del estándar X.509, generalmente la 3, que es la más actual.
- Número de serie: número identificador del certificado, único para cada certificado expedido por una AC determinada.
- Algoritmo de firma: algoritmo criptográfico usado para la firma digital.
- Autoridad certificadora: datos sobre la autoridad que expide el certificado.
- Fechas de inicio y de fin de validez del certificado. Definen el periodo de validez del mismo, que generalmente es de un año.
- Propietario: persona o entidad vinculada al certificado. Dentro de este apartado se usan una serie de abreviaturas para establecer datos de identidad

- Llave pública: representación de la llave pública vinculada a la persona o entidad (en hexadecimal), junto con el algoritmo criptográfico para el que es aplicable.
- Algoritmo usado para la misma para obtener la firma digital de la autoridad certificadora.
- Firma de la autoridad certificadora, que asegura la autenticidad del mismo.
- Información adicional, como tipo de certificado, etc..

Validación de certificados digitales:

Los certificados, debido a su propia naturaleza y al papel que desempeñan, no son documentos imperecederos. En primer lugar, al estar basados en el uso de claves no conviene que sean válidos por períodos largos de tiempo, ya que entre más duración tengan, es más factible que se conozca la clave. Además, cada vez los equipos de computo tienen más recursos para realizar operaciones o cálculos, lo cual facilitaría la labor de los criptoanalistas (que realizan el criptoanálisis). Por este motivo los certificados digitales tienen estipulado un período de validez, que suele ser de un año.

En segundo lugar, es posible que un certificado convenga anularlo en un momento dado, bien sea porque se considere que las claves están comprometidas o porque la persona o entidad propietaria quiebre o sea destituida por haber cometido un delito. Es por esto que existe la posibilidad de revocar o anular un certificado, y esta revocación puede llevarla a cabo el propietario del mismo, la autoridad certificadora o las autoridades judiciales.

Para llevar un control de los certificados revocados (no válidos), las autoridades de certificación han implementado unos servidores especiales que contienen bases de datos en las que figuran los certificados anulados, que se conocen con el nombre de Lista de Certificados Revocados, CRL. Un CRL es un archivo, firmado por la autoridad certificadora, que contiene la fecha de emisión del mismo y una lista de certificados revocados, figurando para cada uno de ellos su número de identificación y la fecha en que ha sido revocado.

Cuando se recibe un certificado digital de otra persona o entidad se comprueba, antes de darlo por válido, si dicho certificado se encuentra en la lista más actualizada de certificados revocados. Si está en la lista, el certificado será rechazado.

Ahora bien, imaginemos que recibimos un certificado como medio de autentificación en una transacción, nuestro sistema comprueba que no está revocado en la última CRL y lo da por válido, pero resulta que al día siguiente aparece como revocado en la CRL nueva. En estos casos deberemos poder demostrar de algún modo que hemos recibido el certificado antes de que se produjera la actualización.

Para solucionar este tipo de situaciones existen los documentos digitales denominados recibos. Un recibo es un documento firmado digitalmente por una persona o entidad de confianza, llamada Autoridad de Oficialía de Partes, que añade la fecha actual a los documentos que recibe para su certificación, firmando luego el resultado con su llave privada. De esta forma los usuarios disponen de un documento que atestigua la hora y fecha exacta en la que se envía o recibe un certificado digital u otro documento electrónico cualquiera.

El uso de un CRL en un proceso de autentificación que presenta varios problemas adicionales. En primer lugar, sólo podemos considerarlo válido cuando la fecha del mismo es igual o posterior a la que queremos usar como referencia en la validez del documento y, en segundo lugar, también puede resultar inadecuado en aquellas operaciones que exijan una velocidad alta en la transacción, sobre todo si el CRL a consultar tiene un tamaño muy grande.

La solución a estos problemas la dan los servicios de directorios o de consulta de certificados, servicios ofrecidos por personas o entidades de confianza aceptada, por el que al recibir una petición de validez de un certificado responde al instante si en esa fecha y hora concreta el mismo es válido o si por el contrario está revocado, en cuyo caso proporcionará también la fecha de revocación. Para dar validez a la respuesta, el servicio de directorios firma con su llave privada la misma, con lo que el usuario estará seguro de la autenticidad de la respuesta recibida.

Protocolos criptográficos

Dentro de la transmisión segura, por medio de certificados digitales, se pueden usar distintos protocolos. A continuación describiremos los protocolos más comunes:

SSL (*Secure Socket Layer*): creado por Nestcape, SSL es un protocolo criptográfico de propósito general que asegura canales de comunicaciones bidireccionales. SSL se utiliza comúnmente junto con el protocolo TCP/IP. Las conexiones de SSL por lo general las inicia un navegador utilizando un prefijo especial en la URL. Por ejemplo: el prefijo http, se utiliza para indicar una conexión de http encriptada con SSL.

SSL ofrece confidencialidad mediante algoritmos de encriptación especificados por el usuario, integridad mediante funciones *hash* criptográficas especificadas por el usuario, y no repudio, mediante mensajes firmados criptográficamente.

¿Sabías que?

Cuando ingresas a un sitio Web que utiliza SSL (https) aparece en la parte inferior de tu navegador un candado que indica que la conexión es segura.

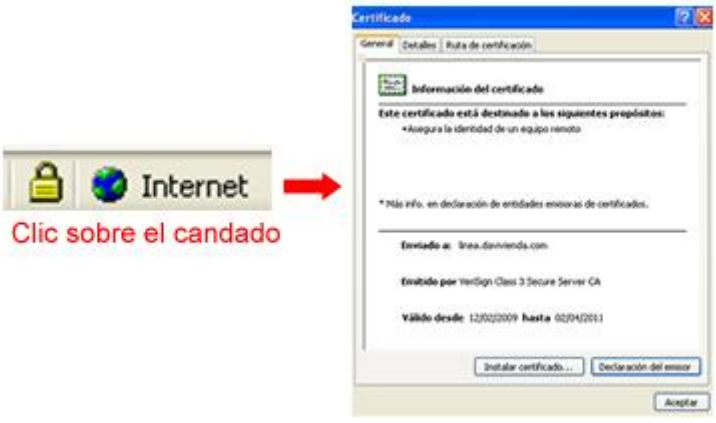

Figura 6: SSL y los certificados digitales

Si le das clic sobre este candado puedes observar el certificado digital del sitio.

En el año 2001 la empresa IETF (*Internet Engineering Task Force*) adquirió los derechos sobre el protocolo SSL (*Secure Socket Layers*), y a partir de ese momento nació TLS (*Transport Layer Security*), este nuevo protocolo buscaba corregir las deficiencias observadas en la última versión de SSL (SSL V3).

El resultado de esta búsqueda fue TLS 1.0, el cual es un protocolo que es compatible en su totalidad con SSL. TLS 1.0 busca integrar en un esquema tipo SSL al sistema operativo, a nivel de la capa TCP/IP, para que el efecto "tunel" que se implementó con SSL sea realmente transparente a las aplicaciones que se están ejecutando. TLS 1.0 tiene varias diferencias con respecto a su antecesor SSL V3, pero las bases del protocolo son las mismas.

SET (*Secure Electronic Transaction*): es un estándar creado conjuntamente por VISA y Master Card. Se trata de un protocolo de seguridad orientado hacia los servicios de pagos electrónicos mediante el uso tarjetas de crédito bancarias. SET proporciona integridad de mensajes, autenticación de datos financieros y encriptación de información sensible.

El protocolo SET usa técnicas criptográficas a fin de ofrecer confidencialidad de la información, asegurar la integridad de los mensajes de pagos y autentificar tanto a los titulares de las tarjetas como a los vendedores. Estas son las características de SET:

I. Ofrece confidencialidad de la información de los medios de pago así como de la información de los pedidos mediante el cifrado de todos los mensajes intercambiados usando algoritmos de clave simétrica y asimétrica.

II. Garantiza la integridad de todos los datos transmitidos gracias al uso de firmas digitales.

III. Ofrece autentificación de que el poseedor de la tarjeta es un usuario legítimo de una cuenta asociada a dicha tarjeta de pago, usando firmas digitales y el certificado del titular de la tarjeta.

IV. Ofrece autentificación de que un vendedor puede aceptar transacciones con tarjetas de pago gracias a su relación con una institución financiera, usando para ello firmas digitales y el certificado del vendedor.

V. Utiliza un protocolo que no depende de otros mecanismos de seguridad de transporte, así como tampoco evita su utilización.

Correo seguro

A comienzos de los años 90 hacen su aparición dos sistemas de correo electrónico seguro: PEM (*Private Enhanced Mail*) y PGP (*Pretty Good Privacy*), a continuación mencionaremos los aspectos generales de cada uno de estos.

PEM (*Private Enhanced Mail*): es un protocolo que utiliza el MD5, RSA y algoritmos IDEA para el cifrado de datos y la comprobación de integridad. PEM contempla la autenticación del origen, confidencialidad, integridad del mensaje, no repudio del origen cuando se utiliza gestión de clave con algoritmo de clave asimétrica. Pero no contempla el control de acceso, la confidencialidad del tráfico de mensajes y el no repudio del mensaje por parte del receptor.

PGP (*Pretty Good Privacity*): es un sistema de criptografía híbrido que usa una combinación de funciones tomadas de la criptografía de clave pública y de la

criptografía simétrica. La orientación principal de PGP es el cifrado y la firma del correo electrónico.

PGP, en su versión 2.6.3i (internacional) se convirtió a mediados de la década de los 90 en un estándar universal. Para su funcionamiento utiliza los siguientes algoritmos:

- IDEA para cifrar con sistema de clave secreta.
- RSA para intercambio de claves y firma digital.
- MD5 para obtener la función *hash* de la firma digital.

Para poder enviar y recibir correo seguro, PGP debe contar con al menos la clave pública del destinatario y el par de claves asimétricas del emisor.

Recomendaciones de cifrado en el desarrollo del software

Al momento de diseñar y desarrollar un sistema, usted debe identificar cuáles son los elementos que necesitan utilizar alguna aplicación criptográfica. Todos los elementos que viajan por la red (interna o externa) que contengan datos o información confidencial deben ser cifrados. Por ejemplo, si usted despliega una página Web que solicita usuario y contraseña, estos datos no deben enviarse de manera plana (sin cifrado) por la red; solo piense en todo lo que podría hacer un intruso una vez conozca ese usuario y contraseña.

Una vez haya identificado los elementos que necesitan utilizar aplicaciones de criptografía o cifrado, se debe escoger el algoritmo o protocolo o el sistema criptográfico que utilizará. Para esto, tenga en cuenta revisar la documentación (páginas, blogs, foros) del producto, busque las debilidades o los factores de riesgos, las características y lógicamente cómo es su aplicación a lo que necesita.

Después de haber implementado el cifrado o la criptografía sobre los elementos de la aplicación, se sugiere hacer las pruebas respectivas. Se recomienda revisar el documento publicado por OWAPS titulado: "*Testing for configuration management*" y realizar las pruebas que apliquen a su aplicación.

A continuación se darán algunas recomendaciones generales relacionadas con la criptografía o cifrado de la información en el diseño y desarrollo del software:

- **Identificación del algoritmo.** Escoja un algoritmo de cifrado que no sea débil. Como tal no hay certeza 100% de que un algoritmo que hoy en día sea infalible, en un futuro continúe siéndolo; por tal razón, se siguiere que al momento de realizar el diseño de la aplicación, se definan los elementos de

tal manera que si debe cambiar el algoritmo en algún momento, no le resulte tan traumático.

Por otro lado, OWAPS define dentro de sus recomendaciones no utilizar los siguientes algoritmos para cifrar datos.

- MD5: este algoritmo es poco seguro porque puede producir el mismo *hash* para dos mensajes distintos, también es poco resistente a ataques de fuerza bruta.
- SHA-0 no debería usarse en ninguna aplicación con datos sensibles ya que es vulnerable.
- SHA-1 es uno de los algoritmos más usados para cifrar, pero OWAPS recomienda utilizar mejor SHA-256 porque implementa un tamaño de clave más largo.
- DES fue una alguna vez el algoritmo criptográfico estándar; pero en la actualidad hasta un PC de escritorio puede romperlo. Hoy en día se prefiere usar AES.

En general, se deben seguir las siguientes instrucciones al momento de escoger algoritmos de encriptación:

Para algoritmos simétricos:

- Un tamaño de clave de 128 bits es suficiente para la mayoría de aplicaciones.
- Considere 168 o 256 bits para sistemas que realicen grandes transacciones financieras.

Para algoritmos asimétricos:

No use tamaños de clave excesivos a menos que sepa que los necesitará, recuerde que entre más grande sea la clave, mayor es el procesamiento de la maquina. Tenga en cuenta:

- Para la mayoría de aplicaciones personales use claves de 1280 bits.
- Para aplicaciones seguras es aceptable una clave de 1536 bits.
- Para aplicaciones altamente protegidas se debe usar claves de 2048 bits.

Hashes:

- Los tamaños de *hash* de 128 bits son suficientes para la mayoría de aplicaciones.
- Considere 168 o 256 bits para sistemas altamente seguros.

Almacenamiento de claves. La criptografía simétrica y asimétrica se basan en el manejo de claves. Como tal, el sistema o plataforma donde quedan guardadas todas las claves debe estar protegido.

A continuación se muestran algunas recomendaciones del manejo de almacenamiento de claves:

- Las claves no se deben almacenar en texto plano, se recomienda guardarla en código cifrado.

- Las claves cifradas deben protegerse en lo posible con permisos del sistema de archivo. Deben ser de solo lectura y solo el usuario o aplicación que accede directamente a ellas puede tener esos permisos.

- Las aplicaciones deben registrar cualquier cambio en las claves. Es decir, cada vez que se cambié una clave, quede consignado quién, cuándo, desde dónde (IP) se hizo el cambio de clave.

- Una forma de proteger un sistema que requiera máxima seguridad consiste en dividir la clave administrativa (todos los permisos) en dos componentes; de tal manera que se requiera dos personas para acceder con este rol al sistema. Cuando se necesite utilizar la clave administrativa las dos personas deberán digitar la parte de la clave que le corresponde, en el orden que se definió al momento de crear la clave; solo así se podrá ingresar a las funciones administrativas del sistema.

- Para algunas aplicaciones Web, es necesario que el servidor Web tenga permisos para acceder a las claves. Si esta es la situación, también es posible que algún código malicioso pueda acceder a las claves. En estos casos, hay que minimizar la funcionalidad de la aplicación para que tenga los permisos estrictos y necesarios sobre las claves; igualmente se debe proteger por todos los medios que estas puedan ser accedidas por un código diferente al de la aplicación Web.

Transmisión Insegura. Cuando se necesita trasmitir datos cifrados a través de la red, es necesario asegurar que el cifrado ocurre antes de que se tramitan los datos a cualquier red no confiable. Tenga en cuenta las siguientes recomendaciones al momento de realizar una transmisión segura:

  - Utilice un algoritmo o protocolo de cifrado adecuado. Si su aplicación necesita un alto grado de seguridad, se recomienda usar una herramienta robusta, por ejemplo: OPENSSL o IPSEC

- Verifique todas las situaciones riesgosas en los puntos de cifrado y descifrado. Si su librería de cifrado necesita usar archivos temporales, pregúntese ¿están protegidos adecuadamente? ¿Están las claves almacenadas de manera segura? ¿Qué ocurre con la información una vez es descifrada, puede ser accedida por externos?

***Tokens* de autenticación**. Las aplicaciones Web tratan con gran número de usuarios. Para diferenciar entre un usuario y otro utilizan *cookies* o identificadores de sesión.

Para la generación de identificadores de sesión los servidores utilizan secuencias predecibles, un atacante sólo necesitaría generar un valor en una secuencia con el fin de presentar un *token* de sesión aparentemente válido. Esto puede ocurrir en gran número de situaciones; a nivel de red para números de secuencia TCP, o bien en la capa de aplicación con las *cookies* usadas como *tokens* de autenticación. La única manera de generar un *token* de autenticación seguro es asegurándose de que no hay manera de predecir su secuencia. En otras palabras: números realmente aleatorios.

La mayoría de los sistemas operativos incluyen funciones para generar números aleatorios que pueden ser llamados desde casi cualquier lenguaje de programación. Por ejemplo: en plataformas Microsoft incluyendo .NET, es recomendable el uso de la función de serie CryptGenRandom. Para todas las plataformas basadas en Unix, OpenSSL es una de las opciones recomendadas ya que ofrece características y funciones API para generar números aleatorios. Para Java utilice java.security.SecureRandom para generar números aleatorios.

Bibliografía

Lucena López .M.J. Criptografía y Seguridad en Computadores. 2002

Michael Smith, THE EMPEROR'S CODES', Bantam Press 2000

Stinson, D. R. Cryptography: Theory and Practice. CRCInc. 1995.

Daemen, "Cipher and Hash Function Design Strategies Based on Linear and Diferential Cryptanalysis," Doctoral Dissertartion, Marzo 1995.

Brunce Schneider , Applied Crytography. 1994

http://computacion.cs.cinvestav.mx/~jjangel/aes/AES_v2005_jjaa.pdf

http://es.wikipedia.org/wiki/Criptograf%C3%ADa

http://es.wikipedia.org/wiki/Data_Encryption_Standard

http://www.scribd.com/doc/6802068/Cap-5-Seguridad-Comunicaciones

http://www.certicamara.com/index.php?option=com_content&task=category&sectionid=13

http://www.ksitdigital.com/productos/api

http://blogs.msdn.com/michael_howard/archive/2005/01/14/353379.aspx