**Trabajo de redes**

08

**Firma digital**

**Inma Gómez Durán**

Índice

¿Qué es firma digital ?.......................................................................3

Funcionamiento firma digital ……………………………………...4

Firmas con clave pública y privada ………………………………..5

Algoritmo Hash ……………………………………………………5

 MD5 ………………………………………………………...6

 SHA-1 ……………………………………………………….7

Programas populares ………………………………………………8

 OpenSSL ……………………………………………………8

 Gnupg ……………………………………………………….8

Administración de claves públicas ………………………………...9

 Certificado …………………………………………………..9

 Estándar X.509 ……………………………………………...9

¿Cómo obtener el dispositivo de firma digital? …………………..11

Bibliografía …………………………………………………….....12

## ¿Qué es firma digital?

 La autenticidad de algunos documentos legales y en general, cualquier tipo de documento se determina mediante el uso de la firma manuscrita ya que ni siquiera sirve una fotocopia de la misma. Para que los documentos enviados de forma digital tengan la misma validez que un documento firmado a mano se crea la firma digital.

Es un método criptográfico que asocia una identidad ya sea de una persona en particular o de un equipo a un mensaje enviado a través de red de transmisión. Su uso puede ser diferente dependiendo de lo que queramos hacer con la firma ya que tendremos posibilidad de validar que el documento es emitido por nosotros, expresar conformidad con algún documento de tipo legal como podría ser la firma de un contrato laboral e incluso asegurar que no podrá modificarse el contenido del mensaje.

En resumen la firma digital nos permitirá tener más seguridad a la hora de emitir un documento de manera íntegra a través de sistemas telemáticos, generalmente la red.

Con la criptografía conseguimos obtener información confidencial que sólo podrán ser entendidas por aquellas identidades que estén autorizadas para recibirlas.

Debe cumplirse pues:

-El receptor debe ser capaz de verificar la identidad del sujeto transmisor.

-El transmisor no puede repudiar el contenido del mensaje que ha enviado.

-El receptor no deber ser capaz de poder construir el mensaje el mismo.

Estas tres normas necesarios podríamos verlas por ejemplo en el caso de un banco donde se debe cumplir la primera regla donde un usuario accede a un cajero automático donde el banco tiene que asegurarse que es su cliente; la segunda norma se aplicaría para proteger al banco del fraude donde un cliente si pide una operación a realizar luego no puede negarla, por ejemplo sacar dinero luego no puede decir que el no ha pedido que se lo descuenten de su cuenta; la tercera se aplica para proteger al cliente en el caso de que el banco trate de falsificar un mensaje firmado por el cliente donde por ejemplo solicite una operación de sustracción de un importe y el banco diga otra.

## Funcionamiento de la firma digital

En general la firma digital está compuesta por una clave pública y otra privada, donde la clave pública se suele distribuir por una autoridad de certificación que es la entidad en la que todos confían, receptor y emisor, y genera el certificado digital.



El esquema de funcionamiento sería el siguiente:

1. El emisor genera un resumen del documento ya que la firma no se realiza sobre el documento completo sino en un resumen del mismo o hash.
2. El emisor firma su resumen encriptándolo con la clave privada propia.
3. El emisor envía el documento y su resumen firmado al receptor.
4. El receptor genera también un resumen del documento que ha recibido usando la misma función que el emisor. Al mismo tiempo descifra el resumen recibido con la clave pública que el emisor ha publicado. Si los resúmenes coinciden la firma será validada.

En este proceso se asegura la autenticidad de que la información es emitida por quien dice, integridad donde asegura que la información no ha sido alterada desde que se envía hasta que llegue al receptor, no repudio donde no se puede negar la autoridad que ha certificado el proceso.

## Firmas con clave pública y simétrica

Si utilizamos firma con clave simétrica existe una autoridad central que posee toda la información de comunicación e intercambio de mensajes y en cual todos los usuarios que la utilizan confían. En este caso el usuario elige su propia clave secreta y la comunica a la autoridad central correspondiente de manera confidencial. Cuando el emisor quiere enviar su mensaje lo genera Ka (B, Ra, t, P) donde B es la identidad del receptor que también debe ser conocido por la entidad central, Ra es un número aleatorio elegido por el emisor, t la marca de tiempo del mensaje y Ka es el mensaje encriptado con la clave del emisor a. Entonces la autoridad sabe que el mensaje es del emisor a, lo desencripta, y envía un mensaje al receptor que contiene el texto generado por el emisor y también el mensaje firmado por la entidad.

Problema: surge cuando una tercera persona es capaz de repetir cualquiera de los dos mensajes donde para minimizar esto se usan marcas de tiempo, así la entidad podría ser capaz de revisar los mensajes más recientes para ver si su usó Ra, si fuera así el mensaje es descartado así como los mensajes que tengan marcas de tiempo muy antiguas. Otro problema es el hecho de que todos tienen que confiar en la autoridad central y estas instituciones no inspiran confianza para todos los usuarios, por tanto sería bueno si la firma de documentos no requiere una autoridad confiable.

La firma con clave pública utiliza algoritmos públicos de encriptación y desencriptación donde E(D(P)) = P además de la propiedad normal de D(E(P)) = P. Así el emisor podrá enviar el mensaje transmitiendo Eb(Da(P)), donde el emisor además de su clave privada (y publica, por supuesto) conoce la clave publica del receptor. Cuando el receptor recibe el mensaje lo transforma usando su clave privada y lo descifra usando la clave pública del emisor (que es conocida).

## Algoritmo hash

Se basa en criptografía simétrica y por el hecho de que el tamaño del documento enviado puede ser grande dificulta el proceso de cifrado por eso se realiza el resumen o hash antes mencionado normalmente con un tamaño fijo. También es necesario saber la fecha en la que los mensajes son enviados para ello se utiliza un “time stamping”. Es relevante su facilidad de cálculo y la colisión de los mensajes ya sea débil (partimos de un mensaje M y usando el algoritmo no podremos encontrar otro mensaje M’ con el mismo valor hash) o fuerte (donde será computacionalmente imposible encontrarlo).

## *MD5*

Es el algoritmo más usado en la actualidad donde el tamaño del mensaje tiene que ser siempre múltiplo de 512, la función de relleno es poner un 1 y todos los 0 que sean necesarios, la longitud real del mensaje se almacena en los 64 bits últimos.



Se dispone de un buffer representado de 128 bits que es inicializado con un valor constante y puede verse como la unión de cuatro registros de 32 dígitos en hexadecimal cada uno de ellos que podríamos llamar A, B, C, D y sus valores son:

A=67452301; B=EFCDAB89; C=98BADCFE; D=10325476

Se realizan rondas de repetición de aplicación del algoritmo donde coge de entrada bloques de 512 bits y los mezcla con los bits que hay en el buffer. Esta operación se repetirá hasta que no queden mas bloques de entrada que consumir y el valor que haya en el buffer una vez acabado el proceso será el mensaje a enviar. El resumen del mensaje es la salida producida por A, B, C y D donde se empieza por el byte de menor peso de A y acaba con el de mayor peso de D.

Aunque nació como un algoritmo criptográficamente seguro el tamaño del hash es suficientemente pequeño para poder ser atacado mediante ataques de ‘fuerza bruta’ y por eso se llevo al descubrimiento de otros algoritmos como el sha-1.

Uno de los usos más importantes y relevantes del algoritmo es en la descarga de archivos de internet que para comprobar que este paquete de descarga no se ha modificado se realiza una comparación con la suma MD5 publicada por los desarrolladores del software. Esto nos puede ayudar mucho en la ayuda contra los ‘troyanos’.

## *SHA-1*

El primer algoritmo llamado SHA apareció publicado en 1993 pero actualmente es llamado SHA-0 ya que posteriormente aparecieron otras versiones del mismo. El SHA-1 apareció en 1995 y usaba un resumen de 160 bits a partir de bloques de 512 bits del mensaje original.

Con esta familia de algoritmos se consigue reducir la complejidad algorítmica ya que se pasa 280 a 264  debido a que se pasa a tener un resumen de 160 bits en lugar del que teníamos con el algoritmo MD5.

Funcionamiento: se empieza rellenando el mensaje con un bit al final seguido de tantos bits 0 como sean necesarios para que la longitud cumpla la condición de tener una longitud múltiple de 512 bits. Después al número de 64 bits que tiene la longitud del mensaje antes del rellenado se le aplica una OR dentro de los 64 bits de menor peso. Durante el cálculo mantiene variables de 32 bits, H0a H4 en las cuales se acumula el hash. Después se procesa cada uno de los bloques M0 a Mn-1. Para el bloque actual, las 16 palabras se copian al inicio de un arreglo auxiliar de 80 palabras, W, después las otras 64 palabras de W se rellenan usando:

Wi = S(Wi-3 XOR Wi-8 XOR Wi-14 XOR Wi-16 con 16<=I <= 79

Después de procesar los primeros bloques de 512 bits, el arreglo W se reinicia pero H se queda como estaba. Cuando acaba este bloque, se inicia el siguiente y así sucesivamente hasta que todos los bloques de 52 bits han sido procesados. Al acabar el último bloque las 5 palabras de 32 bits en el arreglo H se envían a la salida como el hash criptográfico de 160 bits.

En la actualidad este algoritmo parece criptográficamente roto aunque no se conocen los detalles del ataque sobre el mismo realizado por un grupo de prestigiosos investigadores. Recientemente se publico un ataque realizado por tres investigadores chinos que además lo realizaron 64 veces más rápido que los ataques conseguidos anteriormente.

Visto estos ataques el NIST (National institute of standards and technology) da como alternativa al uso de este algoritmo la utilización de estándares de mayor longitud como: SHA-224, SHA-256, SHA-384, SHA-512 y RIPEMD-160 que fue desarrollado abiertamente y también tiene versiones de 256 y 320 bits.

## Programas populares

Vamos a ver una pequeña referencia a los programas de generación de certificados más usados comúnmente.

## *Openssl*

Es un proyecto software de libre descarga creado por la comunidad Open Source y basado en SSLeay, y consiste en un paquete de herramientas de administración y librerías que suministran funciones criptográficas a otros paquetes y navegadores además de ayudar al sistema en la implementación del SSL y otros protocolos relacionados con la seguridad.

Lo podríamos descargar de la siguiente pagina web: “http://openssl.softonic.com/linux/descargar”

## *Gnupg*

Es un sistema de claves públicas donde cada usuario tendrá una clave privada y otra pública donde la primera será usada para desencriptar el mensaje que nos fue encriptado con la clave pública.

Para la generación de las claves se pide una serie de parámetros como el tipo de clave, su tamaño que suele variar entre 1024 y 4096 bits aunque por defecto suele ser 2048, también serán importantes los parámetros de periodicidad y validez de las claves.

**Lo podremos descargar de:** [**http://www.gnupg.org**](http://www.gnupg.org)

## Administración de claves públicas

## El uso de la clave publica permite que personas que no disponen de una clave pública para compartir se comuniquen entre ellos y que puedan firmar digitalmente documentos sin tener una tercera entidad confiable entre ellos. Sin embargo surge un problema: el emisor y receptor no se conocen y por tanto no poseen las claves públicas el uno del otro, así pues habrá que establecer una solución donde se utilice un mecanismo para asegurar que las claves públicas puedan intercambiarse de forma segura.

##

## *Certificado*

Existe una organización que certifica claves públicas de personas, empresas y organizaciones que se llama autoridad de certificación (CA). Un certificado tiene que ser capaz de enlazar una clave pública junto al nombre del titular ya sea una empresa o una persona propia aunque también se puede enlazar a un atributo. Los certificados en si no están protegidos ni siquiera son secretos.

Si el certificado es autentico y la autoridad CA es de nuestra confianza podemos confiar en que el sujeto titular del certificado tiene la llave publica que se afirma en dicho documento. Los certificados también sirven para identificar usuarios en internet y así se pueden configurar servidores capaces de filtrar el acceso a dichos usuarios.

## *Estándar X.509*

Es una de las propuestas más antiguas para una infraestructura de clave pública y es una forma de describir certificados.

Se describe por una serie de campos:

Versión: identifica la versión que se utiliza.

Numero de serie: junto con el nombre de la CA identifican de manera univoca el certificado.

Algoritmo de firma: algoritmo que se uso para realizar la firma del certificado.

Emisor: el nombre X.500 de la CA.

Validez: tiempo comprendido entre un intervalo de fechas donde el certificado tiene validez.

Nombre del sujeto: entidad cuya clave se certifica.

Clave pública: ID del algoritmo usado para generar la clave pública y ella misma.

ID del emisor: ID opcional que identifica de forma univoca al emisor del certificado.

ID del sujeto: ID opcional que identifica de manera univoca al sujeto del certificado.

Firma: generada con la clave privada de CA.

Extensiones: diferentes extensiones definidas.

Este estándar tiene tres procedimientos alternativos para la autenticación en peticiones de servicio o envío de información:

-Autenticación a una vía: el mensaje está formado por la marca de tiempo y el testigo, además también puede estar acompañado de una clave de la sesión temporal entre A y B. Con esto se define que el emisor es A y el mensaje fue enviado a B de manera íntegra y única.



-Autenticación a dos vías: la información además de enviarse del emisor al receptor también lo hace de manera inversa, así que añade al anterior procedimiento que el receptor también es capaz de generar un mensaje dirigido hacia el emisor de manera íntegra.



-Autenticación a tres vías: se utiliza cuando entre el emisor y el receptor no existen o tiene los relojes sincronizados, además de realizar la autenticación a dos vías el emisor envía una respuesta a la respuesta del testigo incluyendo el nuevo testigo contenido en la respuesta original.



## ¿Cómo obtener el dispositivo para firmar digitalmente el mensaje?

El proceso de firmado sería el siguiente:

1. Solicitar el par de claves y certificado digital a la entidad Prestadora de Servicios capaces de Certificación ya sea por internet o personándose en la oficina.
2. La entidad será responsable de verificar que soy la persona que digo ser mediante la muestra del DNI.
3. La entidad genera el par de claves y el certificado digital que corresponde a esas claves.
4. Es entregada por la entidad una tarjeta magnética donde se han grabado previamente el par de claves y el certificado de manera protegida como se suele hacer en las tarjetas bancarias, también puede entregarse en lugar de la tarjeta el certificado en su propia página web para que el usuario lo instale directamente en su PC.
5. Con un lector de tarjetas adecuado conectado a un PC y esta tarjeta magnética que nos han entregado podremos firmar digitalmente los mensajes electrónicos que queramos enviar.



# BIBLIOGRAFIA

He consultado las siguientes referencias:

- www.tuGuíaLegal.com / La Firma Digital.mht

-redes de computadores 4º edición Andrew S.Tanenbaum

-http://www.alu.ua.es/f/fgc10/CERTIFICADOS%20X.509.ppt

-http://www.programacion.net/ y luego buscas en [**Inicio**](http://www.programacion.net/) **>** [**Artículos**](http://www.programacion.net/articulos/) **>** [**Internet**](http://www.programacion.net/articulos/internet/) **>** [**PHP**](http://www.programacion.net/articulos/php/) **> Enviando passwords de forma segura con MD5 y PHP3.**

**-**http://es.wikipedia.org/wiki/Portada de aquí he buscado muchas definiciones.

-http://www.iec.csic.es/CRIPTONOMICON/correo/firma.html

-http://www.fd.com.ar/manual.htm#FIRMA

-http://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/md5.html

-http://mundocripto.com/mambo//content/category/12/39/65/