Aplicaciones de la criptografía

La criptografía es una disciplina con multitud de aplicaciones, muchas de las cuales están en uso hoy en día. Entre las más importantes destacamos las siguientes:

- **Seguridad de las comunicaciones**. Es la principal aplicación de la criptografía a las redes de computadores, ya que permiten establecer canales seguros sobre redes que no lo son. Además, con la potencia de cálculo actual y empleando algoritmos de cifrado simétrico (que se intercambian usando algoritmos de clave pública) se consigue la privacidad sin perder velocidad en la transferencia.
- **Identificación y autentificación**. Gracias al uso de firmas digitales y otras técnicas criptográficas es posible identificar a un individuo o validar el acceso a un recurso en un entorno de red con más garantías que con los sistemas de usuario y clave tradicionales.
- **Certificación**. La certificación es un esquema mediante el cual agentes fiables (como una entidad certificadora) validan la identidad de agentes desconocidos (como usuarios reales). El sistema de certificación es la extensión lógica del uso de la criptografía para identificar y autentificar cuando se emplea a gran escala.
- **Comercio electrónico**. Gracias al empleo de canales seguros y a los mecanismos de identificación se posibilita el comercio electrónico, ya que tanto las empresas como los usuarios tienen garantías de que las operaciones no pueden ser espiadas, reduciéndose el riesgo de fraudes y robos.

Algoritmos de cifrado simétrico

DES

El DES (*Data Encription Standard* o *Estándar de Encriptación de Datos*) es el nombre del documento FIPS (Federal Information Processing Standard) 46-1 del Instituto Nacional de Estándares y Tecnología (NIST) del Departamento de Comercio de Estados Unidos. Fue publicado en 1977. En este documento se describe el DEA (*Data Encription Algorithm* o *Algoritmo de Encriptación de Datos*. Es el algoritmo de cifrado simétrico más estudiado, mejor conocido y más empleado del mundo.

El DEA (llamado con frecuencia DES) es un algoritmo de cifrado por bloques de 64 bits de tamaño. Emplea una clave de 56 bits durante la ejecución (se eliminan 8 bits de paridad del bloque de 64). El algoritmo fue diseñado para ser implementado en hardware. Cuando se utiliza en comunicaciones ambos participantes deben conocer la clave secreta (para intercambiarla se suelen emplear algoritmos de clave pública). El algoritmo se puede usar para encriptar y desencriptar mensajes, generar y verificar códigos de autentificación de mensajes (MAC) y para encriptación de un sólo usuario (p. ej para guardar un archivo en disco).

Aunque el DES era un algoritmo computacionalmente seguro, esto ha dejado de ser cierto, ya que con hardware específico es posible realizar ataques por fuerza bruta que descubran una clave en pocos días . El problema principal es que el tamaño de la clave (56 bits) es demasiado pequeño para la potencia de cálculo actual. De hecho, el DES dejó de ser el algoritmo empleado por el gobierno norteamericano en Noviembre de 1998 y de momento (hasta que el AES sea elegido), emplean el Triple DES.

Triple-DES

Consiste en encriptar tres veces una clave DES. Esto se puede hacer de varias maneras:

- DES-EEE3: Tres encriptaciones DES con tres claves distintas.
- DES-EDE3: Tres operaciones DES con la secuencia encriptar-desencriptar-encriptar con tres claves diferentes.
- DES-EEE2 y DES-EDE2: Igual que los anteriores pero la primera y tercera operación emplean la misma clave.

Dependiendo del método elegido, el grado de seguridad varía; el método más seguro es el DES-EEE3.

AES

El AES (*Advanced Encription Standard* o *Estándar Criptográfico Avanzado*) es un algoritmo de cifrado por bloques destinado a reemplazar al DES como estándar.

Algoritmos de clave pública

RSA

El RSA, llamado así por las siglas de sus creadores (*Rivest, Shamir y Adelman*), es el algoritmo de clave pública más popular. El algoritmo se puede usar para encriptar comunicaciones, firmas digitales e intercambio de claves.

La clave es de tamaño variable, generalmente se usan claves entre 512 y 2048 bits. Las claves más grandes aumentan la seguridad del algoritmo pero disminuyen su eficiencia y generan más texto cifrado. Los bloques de texto en claro pueden ser de cualquier tamaño, siempre que sea menor que la longitud de la clave. Los bloques de texto cifrado generados son del tamaño de la clave.

La *clave pública* del algoritmo tiene la forma (e, n), donde e es el exponente y n el módulo. La longitud de la clave es igual al número de bits de n. El *módulo* se obtiene multiplicando dos números primos grandes, p y q. Los números se seleccionan aleatoriamente y se guardan en secreto. La *clave privada* tiene la forma (d, n), donde d es el producto inverso de e modulo (p-1)(q-1) (es decir, (ed - 1) es divisible por (p-1)(q-1)).

El cálculo de d a partir de p y q es sencillo, pero es computacionalmente imposible calcular d sin conocer p y q para valores grandes de n, ya que obtener sus valores es equivalente a factorizar n, que es un problema intratable computacionalmente.

El funcionamiento del algoritmo es como sigue:

- **Encriptación**.Para encriptar un mensaje un usuario calcula $c=m^e$ modulo n, donde m es el texto en claro, c es el texto cifrado y (e, n) es la clave pública del destinatario.
- **Desencriptación**. Para desencriptar el mensaje el destinatario calcula c^d modulo n = $(m^e)^d$ modulo n = m^{ed} modulo n = m, donde (d, n) es la clave privada del destinatario. Hay que indicar que la última sustitución es posible por el modo en que hemos escogido los números, ya que d es el producto inverso de e modulo n, por lo que m^{ed}= m.
- **Firmado**. Si el emisor desea enviar el mensaje firmado usa su clave privada para calcular $c=m^d$ modulo n y el destinatario lo valida calculando c^e modulo n = $(m^d)^e$ modulo n = m^{de} modulo n = m, donde (e, n) es la clave pública del emisor.

El algoritmo es lento, ya que emplea operaciones matemáticas que tienen un coste elevado y trabaja con claves de gran tamaño.

Comparado con los sistemas de cifrado simétrico como el DES, el algoritmo de RSA es 100 veces más lento en software y de 1000 a 10000 veces más lento en hardware.

Funciones de dispersión (algoritmos HASH)

SHA y SHA-1

El SHA (*Secure Hash Algorithm*) es un algoritmo de resumen seguro desarrollado por el NIST. El SHA-1 es una versión corregida del algoritmo publicada en 1994. El algoritmo es un estándar ANSI.

El algoritmo toma un mensaje de menos de 2^{64} bits y genera un resumen de 160 bits. Es más lento que el MD5, pero la mayor longitud de clave lo hace más resistente a ataques de colisión por fuerza bruta y de inversión.

MD2, MD4 y MD5

Los tres son algoritmos de resumen de mensajes (el MD viene de *Message Digest*) desarrollados por Rivest.

Los tres toman un mensaje de longitud arbitraria y generan un resumen de 128 bits. El MD2 está optimizado para máquinas de 8 bits, mientras que el MD4 y MD5 son para arquitecturas de 32 bits.

Firmas digitales

DSA y DSS

El DSA (*Digital Signature Algorithm* o *Algoritmo Estándar de Firmado*) es el algoritmo de firmado digital incluido en el DSS (*Digital Signature Standard* o *Estándar de Firmas Digitales*) del NIST Norteamericano. Se publicó en 1994.

El DSA está basado en el problema de los logaritmos discretos y sólo puede emplearse para las firmas digitales (a diferencia del RSA, que también puede emplearse para encriptar). La elección de este algoritmo como estándar de firmado generó multitud de críticas: se pierde flexibilidad respecto al RSA (que además, ya era un estándar *de hecho*), la verificación de firmas es lenta, el proceso de elección fue poco claro y la versión original empleaba claves que lo hacían poco seguro.

El algoritmo es más rápido para generar la firma que para validarla, al revés de lo que sucede con el RSA.

Certificados Digitales

Un *certificado de clave pública* es un punto de unión entre la clave pública de una entidad y uno o más atributos referidos a su identidad. El certificado garantiza que la clave pública pertenece a la entidad identificada y que la entidad posee la correspondiente clave privada.

Los *certificados de clave pública* se denominan comúnmente *Certificado Digital*, *ID Digital* o simplemente *certificado*. La entidad identificada se denomina *sujeto del certificado* o *subscriptor* (si es una entidad legal como, por ejemplo, una persona).

Los certificados digitales sólo son útiles si existe alguna *Autoridad Certificadora* (*Certification Authority* o *CA*) que los valide, ya que si uno se certifica a sí mismo no hay ninguna garantía de que su identidad sea la que anuncia, y por lo tanto, no debe ser aceptada por un tercero que no lo conozca.

Es importante ser capaz de verificar que una autoridad certificadora ha emitido un certificado y detectar si un certificado no es válido. Para evitar la falsificación de certificados, la entidad certificadora después de autentificar la identidad de un sujeto, firma el certificado digitalmente.

Los *certificados digitales* proporcionan un mecanismo criptográfico para implementar la autentificación; también proporcionan un mecanismo seguro y escalable para distribuir claves públicas en comunidades grandes.

Certificados X.509

El formato de certificados X.509 es un estándar del ITU-T (*International Telecommunication Union-Telecommunication Standarization Sector*) y el ISO/IEC (*International Standards Organization / International Electrotechnical Commission*) que se publicó por primera vez en 1988. El formato de la versión 1 fue extendido en 1993 para incluir dos nuevos campos que permiten soportar el control de acceso a directorios. Después de emplear el X.509 v2 para intentar desarrollar un estándar de correo electrónico seguro, el formato fue revisado para permitir la extensión con campos adicionales, dando lugar al X.509 v3, publicado en 1996.

Los elementos del formato de un certificado X.509 v3 son:

- **Versión**. El campo de versión contiene el número de versión del certificado codificado. Los valores aceptables son 1, 2 y 3.
- **Número de serie del certificado**. Este campo es un entero asignado por la autoridad certificadora. Cada certificado emitido por una CA debe tener un número de serie único.
- **Identificador del algoritmo de firmado**. Este campo identifica el algoritmo empleado para firmar el certificado (como por ejemplo el RSA o el DSA).
- **Nombre del emisor**. Este campo identifica la CA que ha firmado y emitido el certificado.

- **Periodo de validez**. Este campo indica el periodo de tiempo durante el cual el certificado es válido y la CA está obligada a mantener información sobre el estado del mismo. El campo consiste en una fecha inicial, la fecha en la que el certificado empieza a ser válido y la fecha después de la cual el certificado deja de serlo.
- **Nombre del sujeto**. Este campo identifica la identidad cuya clave pública está certificada en el campo siguiente. El nombre debe ser único para cada entidad certificada por una CA dada, aunque puede emitir más de un certificado con el mismo nombre si es para la misma entidad.
- **Información de clave pública del sujeto**. Este campo contiene la clave pública, sus parámetros y el identificador del algoritmo con el que se emplea la clave.
- **Identificador único del emisor**. Este es un campo opcional que permite reutilizar nombres de emisor.
- **Identificador único del sujeto**. Este es un campo opcional que permite reutilizar nombres de sujeto.
- **Extensiones**.

Las extensiones del X.509 v3 proporcionan una manera de asociar información adicional a sujetos, claves públicas, etc.

Autoridades Certificadoras

Una *autoridad certificadora* es una organización fiable que acepta solicitudes de certificados de entidades, las valida, genera certificados y mantiene la información de su estado.

Una CA debe proporcionar una *Declaración de Prácticas de Certificación* (*Certification Practice Statement* o *CPS*) que indique claramente sus políticas y prácticas relativas a la seguridad y mantenimiento de los certificados, la responsabilidades de la CA respecto a los sistemas que emplean sus certificados y las obligaciones de los subscriptores respecto de la misma.

Las labores de un CA son:

- **Admisión de solicitudes**. Un usuario rellena un formulario y lo envía a la CA solicitando un certificado. La generación de las claves pública y privada son responsabilidad del usuario o de un sistema asociado a la CA.
- **Autentificación del sujeto**. Antes de firmar la información proporcionada por el sujeto la CA debe verificar su identidad. Dependiendo del nivel de seguridad deseado y el tipo de certificado se deberán tomar las medidas oportunas para la validación.
- **Generación de certificados**. Después de recibir una solicitud y validar los datos la CA genera el certificado correspondiente y lo firma con su clave privada. Posteriormente lo manda al subscriptor y, opcionalmente, lo envía a un almacén de certificados para su distribución.
- **Distribución de certificados**. La entidad certificadora puede proporcionar un servicio de distribución de certificados para que las aplicaciones tengan acceso y puedan obtener los certificados de sus subscriptores.
- **Anulación de certificados**. Al igual que sucede con las solicitudes de certificados, la CA debe validar el origen y autenticidad de una solicitud de anulación. La CA debe mantener información sobre una anulación durante todo el tiempo de validez del certificado original.
- **Almacenes de datos**. Hoy en día existe una noción formal de *almacén* donde se guardan los certificados y la información de las anulaciones. La designación oficial de una base de datos como almacén tiene por objeto señalar que el trabajo con los certificados es fiable y de confianza.

Los protocolos SSL y TLS

En este apartado discutiremos mecanismos para establecer canales seguros para aplicaciones de red a nivel de la capa de transporte. Trataremos los protocolos SSL y TSL, que son los más utilizados en la actualidad para proporcionar versiones seguras de protocolos de red como el http (https).

Los protocolos SSL y TLS

En este apartadodiscutiremos mecanismos para establecer canales seguros para aplicaciones de red a nivel de la capa de transporte. Trataremos los protocolos SSL y TSL, que son los más utilizados en la actualidad para proporcionar versiones seguras de protocolos de red como el http (https).

El protocolo TLS

El protocolo TLS (*Transport Layer Security*) es una evolución del protocolo SSL (*Secure Sockets Layer*).

Los objetivos del protocolo son varios:

1. **Seguridad criptográfica**. El protocolo se debe emplear para establecer una conexión segura entre dos partes.
2. **Interoperabilidad**. Aplicaciones distintas deben poder intercambiar parámetros criptográficos sin necesidad de que ninguna de las dos conozca el código de la otra.
3. **Extensibilidad**. El protocolo permite la incorporación de nuevos algoritmos criptográficos.
4. **Eficiencia**. Los algoritmos criptográficos son costosos computacionalmente, por lo que el protocolo incluye un esquema de *cache de sesiones* para reducir el número de sesiones que deben inicializarse desde cero (usando criptografía de clave pública).

El protocolo está dividido en dos niveles:

- **Protocolo de registro TLS** (*TLS Record Protocol*).
- **Protocolo de mutuo acuerdo TLS** (*TLS Handshake Protocol*).

El de más bajo nivel es el *Protocolo de Registro*, que se implementa sobre un protocolo de transporte fiable como el TCP. El protocolo proporciona seguridad en la conexión con dos propiedades fundamentales:

1. **La conexión es privada**. Para encriptar los datos se usan algoritmos de cifrado simétrico. Las claves se generan para cada conexión y se basan en un secreto negociado por otro protocolo (como el de mutuo acuerdo). El protocolo también se puede usar sin encriptación.
2. **La conexión es fiable**. El transporte de mensajes incluye una verificación de integridad.

El *protocolo de registro* se emplea para encapsular varios protocolos de más alto nivel, uno de ellos, el *protocolo de mutuo acuerdo*, permite al servidor y al cliente autentificarse mutuamente y negociar un algoritmo de encriptación y sus claves antes de que el protocolo de aplicación transmita o reciba datos.

El *protocolo de mutuo acuerdo* proporciona seguridad en la conexión con tres propiedades básicas:

1. La identidad del interlocutor puede ser autentificada usando criptografía de clave pública. Esta autentificación puede ser opcional, pero generalmente es necesaria al menos para uno de los interlocutores.
2. La negociación de un secreto compartido es segura.
3. La negociación es fiable, nadie puede modificar la negociación sin ser detectado por los interlocutores.

Aplicaciones e implementaciones

El protocolo SSL/TLS tiene multitud de aplicaciones en uso actualmente. La mayoría de ellas son versiones seguras de programas que emplean protocolos que no lo son. Hay versiones seguras de servidores y clientes de protocolos como el http, nntp, ldap, imap, pop3, etc.

Existen multitud de implementaciones del protocolo, tanto comerciales como de libre distribución. Una de las más populares es la biblioteca openssl, escrita en C y disponible bajo licencia GNU. Incluye todas las versiones del SSL y el TLS y un gran número de algoritmos criptográficos, algunos de los cuales ni tan sólo son empleados en el estándar TLS. La biblioteca está disponible en el URL http://www.openssl.org. En esa misma dirección se puede encontrar una lista de referencias a otras implementaciones gratuitas y comerciales de los protocolos SSL y TLS y aplicaciones que los emplean.