Cripto light

Apunte de clase sobre el uso de criptografía en redes. Teoría de las comunicaciones (FCEN, UBA).
Para usar en 2005-11-16. Escrito por Ernesto Alvarez.

Temario

1. Principios
   1.1. Confidencialidad
   1.2. Integridad
   1.3. Autenticación (y las diferencias con autorización)
   1.4. No Repudio
2. Clave pública vs Clave privada
3. Métodos básicos
   3.1. Encripción (aca se explica con ejemplos clave pública vs. privada)
      a) Fundamentos
      b) Ejemplos: RSA, ElGamal, DES/3DES, AES, IDEA, BLOWFISH
      c) The possibility of Secure Non-Secret Digital Encryption (http://www.cesg.gov.uk/site/publications/media/possnse.pdf)
   3.2. Firma digital
   3.3. Hashing
      a) Hashing no criptograficamente seguro
      b) Ataque de cumpleaños
      c) Propiedades
      d) Ejemplos: MD5, SHA-1 (y variantes), CAST, TIGER, RIPEMD-160
   3.4. MAC
   3.5. KEX
      a) Bases y ejemplos: DH
      b) Adversarios activos versus pasivos
4. Métodos avanzados
   4.1. Uso eficiente de PKI
   4.2. Verificación indirecta de identidad
      a) Certificados
      b) PGP y web of trust (en relación a certificados)
5. Temas varios
   5.1. Porque es necesario un buen RNG
   5.2. One time pads y snake oil (http://www.faqs.org/faqs/cryptography-faq/snake-oil/)

Principios

Hay cuatro partes fundamentales en la criptología, objetivos de los sistemas. Estos cuatro objetivos son los que se buscan lograr con las implementaciones de protocolos criptográficos.

Confidencialidad

Es un servicio que tiene como objetivo mantener una información oculta (en el sentido de incompensible) de terceros no autorizados. Por ejemplo, si quisiera mandar un mensaje confidencial a un tercero, puedo encriptarlo para garantizar confidencialidad.

Integridad

Es la habilidad de detectar modificaciones no autorizadas a datos, abarcando inserciones, borrados y alteraciones. Muchas aplicaciones requieren integridad, sin confidencialidad. Por ejemplo registros públicos. Un error comun es pensar en la ciptología como proveedora unicamente de confidencialidad.

Autenticación

Autenticación brinda la posibilidad de determinar que una entidad es quien dice ser. Puede aplicarse tanto a entidades (personas) como a datos. Aplicado a una persona, significa determinar que esta persona es quien dice ser. Aplicado a datos, significa conocer su origen y características.

No debe confundirse con autorización, que significa tener un medio para decidir quien tiene derecho a realizar ciertas acciones y quienes no. Por ejemplo, si alguno de ustedes intentara llevarse parciales para corregir (suponiendo que no intentaran hacerse pasar por un docente), se toparían con un problema de autorización, no existiría dudas de que son alumnos de la cátedra (autenticación exitosa), pero sabemos que no están a cargo de la corrección de parciales (fallo de autorización).

No repudio

Es la propiedad que impide que una entidad niegue a posteriori un mensaje que emitió. Por ejemplo, alguien autorizando una compra y luego negando haber realizado esa transacción.

Clave pública versus clave privada

La mayoría de las operaciones criptográficas tienen dos partes: una que realiza la entidad emisora (por ejemplo agregar información de autenticación a un texto) y una realizada por el receptor (confirmar dicha información). Este tipo de operaciones requieren generalmente una clave. Hay dos grandes familias de algoritmos criptográficos. La más antigua es la de algoritmos de clave privada o clave simétrica, en donde la clave aplicada en la segunda etapa es igual (o trivialmente relacionada) a la clave aplicada en la primera. Este tipo de algoritmos requiere que esta clave sea compartida entre las dos entidades por medio de un canal seguro. Recientemente (mediados de la decada del '70) surgió la segunda familia, los algoritmos de clave pública o asimétrica, en donde las claves a usar en la primera y segunda parte no son las mismas, y son dificiles de asociar, aunque haya una relación entre ellas. Esto permite que una de estas claves (la clave pública) sea distribuída por un canal inseguro sin que ello signifique un quiebre del sistema. Hoy en dia son una parte importante de la seguridad en redes.

Encripción

Definamos un algoritmo de encripción como una función (caja negra) que toma un texto plano y una clave, y devuelve un texto incomprensible (el texto cifrado, ciphertext). Esta función es biyectiva y su inversa es la función de desencripción. Esta función depende de una clave, que es necesaria para obtener el texto plano original. Si el algoritmo es simétrico, es de esperar que la clave usada en ambas etapas sea la misma, mientras que de ser un algoritmo de clave pública, las claves serán distintas. Cuando el algoritmo es simétrico, el uso consiste en intercambiar previamente la clave para que luego cada entidad la aplique cuando sea necesario. Obviamente es necesario proteger este intercambio, puesto que si la clave cae en manos de un oponente, el sistema dejará de ser seguro (incluso grabaciones de mensajes anteriores pueden ser descifradas).

Esto no es tan simple cuando el algoritmo de clave pública. Si se está usando un algoritmo de esta clase, habrán dos claves. En este caso la idea es simple, aunque no tan directa como en el caso simétrico. La entidad que desea recibir los mensajes genera este par de claves. Esta entidad guarda una de ellas (la clave secreta) mientras que distribuye la clave pública a quien la desee (la publica). Una persona que quiere enviarle un mensaje solo debe obtener la clave pública y encriptar el mensaje con ella. El receptor simplemente usa su clave privada para obtener el mensaje original. En este caso el problema no consiste en mantener secreta la clave, sino en asegurarse que el potencial emisor reciba la clave correcta y no una falsificación que un atacante haya distribuído con el fin de interceptar los mensajes de otros.

El estandar anterior de algoritmos de clave privada es DES, que tiene una longitud de clave demasiado corta como para ser seguro, ya que se puede probar todo el espacio de claves en un tiempo razonable (a finales de los '90 tomaba 24 horas pagando USD 100000 en equipos). Una variante es 3DES, que es DES aplicado 3 veces, lo que duplica o triplica el largo de la clave (haciéndolo seguro contra ataques de fuerza bruta).

Puesto que este algoritmo es demasiado lento, se hizo un concurso para elegir el nuevo estandar, AES. Este concurso fue ganado por el algoritmo llamado Rijndael.

También hay otros buenos algoritmos, como IDEA, Blowfish o RC6, algunos también finalistas del concurso AES.

El ejemplo clásico de algoritmo de clave pública es RSA, descubierto por Rivest, Shamir y Adleman. Otro algoritmo es ElGamal, creado por el criptólogo de mismo nombre, relacionado con el estandar de firma DSS.

A pesar de que publicamente los algoritmos de clave pública fueron descubiertos a a mediados de los '70 (RSA en 1978), fueron descubiertos en secreto por el GCHQ, la agencia de inteligencia de señales británica. Entre la serie de papers originales, hay uno extremadamente interesante llamado “The possibility of non-secret encryption” que habla de la idea de la criptografía de clave pública como concepto teórico, antes de que se descubrieran las funciones en sí. Este paper es extremadamente simple de leer, y ofrece una forma muy interesante de expresar el concepto de criptografía de clave pública al mostrarlo como una serie de tablas y permutaciones, y es lectura recomendada para quienes deseen profundizar sobre el tema.

Firma digital

Si miran con cuidado el mecanismo de encripción de clave pública, van a notar un uso normalmente imposible con algoritmos de clave privada. Una entidad podría distribuir su clave pública normalmente, pero en vez de recibir mensajes de terceros, ella misma puede "encriptar" mensajes usando su clave privada. Cualquier otra entidad puede (si tiene la parte pública de la clave) "desencriptar" este mensaje, obteniendo el mensaje original. Cuando el mecanismo se usa al revés, surge el concepto de firma digital. Esto se basa en que la única persona capaz de realizar la operación con la clave privada es el dueño de la misma, por lo que se garantiza que el emisor del mensaje.

Hashing

En algoritmos 2 vieron lo que es una función de hashing, una función que mapea muchos valores a una cantidad limitada de "baldes". Para este momento deberían darse cuenta que un checksum o un CRC son un tipo de funciones de este tipo. Si bien estas funciones son buenas para corregir errores accidentales, no son adecuadas contra un atacante determinado, ya que es trivial realizar la operación inversa y obtener un posible mensaje dado un checksum. También es trivial obtener una serie de modificaciones a realizar a un mensaje para que este sea aceptado como bueno cuando no lo es. Definamos entonces ciertas propiedades para que una función de hashing pueda ser usada para fines criptográficos:

1. No debe ser trivialmente invertible
2. No debe ser "facil" (por facil quiero decir práctico lograrlo en un período razonable), dado un mensaje, encontrar un segundo mensaje con el mismo hashing.
3. No debe ser "facil" encontrar dos mensajes con el mismo hashing.

Aunque 2 y 3 parezcan la misma condición, no lo son. Piensen en el ataque del cumpleaños que debieron ver en Algo 2 (explicar si no lo vieron).

Ejemplos de funciones que cumplen estas propiedades son MD5, SHA-1 (y variantes), CAST, TIGER y RIPEMD-160 entre otras.

MAC

Una función de hashing puede tener propiedades muy interesantes, pero aún con todas ellas, un intruso puede computarlas tan facilmente como lo puede hacer el emisor y receptor, por lo que no son suficientes para garantizar la integridad de un mensaje. Sin embargo, si tanto receptor como emisor tienen una clave secreta compartida, la integridad puede garantizarse realizando una modificación a una función de hashing. Si una función de hashing depende no solo del texto, sino también de una clave, se la puede usar para autenticar un mensaje, ya que el intruso (que no posee la clave correcta) no será capaz de recomputar el hashing de un mensaje que hubiera modificado. Este tipo de hashing con clave se llama "Message Authentication Code" o MAC.

Key Agreement/Key Exchange

Un análisis de los algoritmos de clave pública (como los mencionados arriba) lleva a una pregunta importante: “Es posible usarlos para intercambiar un valor secreto, que puede servir como clave?”. La respuesta es “Si”, y analizando la encripción de clave pública se puede ver que es trivial elegir un valor al azar y enviarlo encriptado al destinatario. A este mecanismo se lo llama un algoritmo de intercambio de claves (Key Exchange). Esta no es la única forma de elegir esta clave: otra variante consiste en la obtención de una clave basándose en la información introducida por AMBAS partes, lo que se conoce como “Key Agreement”. En la materia los vamos a tratar de manera intercambiable. Cabe aclarar que el primer algoritmo de clave pública descubierto (en el ámbito académico) fué el algoritmo Diffie-Hellman, que es un algoritmo en el cual ambas partes contribuyen para crear la clave.

Un punto importante respecto a KEX es que si bien es posible frustrar a un atacante pasivo (que no puede alterar los mensajes), un atacante activo puede colocarse en el medio, haciéndose pasar por el interlocutor del lado opuesto en un ataque llamado “Hombre en el medio”.

Criptografía de clave pública eficiente

Un detalle respecto a la criptografía de clave pública es que es extremadamente lenta en comparación con la de clave simétrica (una relacion de 1000:1). Por ello es deseable minimizar el uso de operaciones públicas y maximizar las operaciónes de clave secreta. La idea de intercambio de claves da una posibilidad de lograr estos objetivos, aunque no siempre se tiene al interlocutor en vivo (por ejemplo cuando se envía un mensaje encriptado). Sin embargo es perfectamente posible enviar un mensaje encriptado y autenticado de forma eficiente dadas las condiciones necesarias para operar con clave pública.

Se tiene la clave pública del receptor y un mensaje a enviar. Si quisiera enviar un mensaje encriptado, lo unico que debo hacer es computar una clave al azar, llamada clave de sesión, encriptar simétricamente el mensaje con la clave de sesión y luego encriptar esta clave con la clave pública del destinatario. Esto hace que el texto encriptado por el algoritmo de clave pública sea pequeño en relación al texto original.

Para firmar un mensaje, la idea es similar. Se toma el texto original, se aplica una función de hashing y se firma este hashing con la clave privada propia. Estas dos estrategias pueden aplicarse una sobre otra, lo que permite un texto encriptado y firmado.

Verificación indirecta de identidad

Es perfectamente factible querer establecer una comunicación con alguien a quien nunca se conoció de antemano. En ese caso no hay forma de verificar directamente que una clave corresponda al interlocutor y no a un impostor. En esos casos usualmente se recurre a un tercero o terceros en el cual ambos confían y que verificó la identidad de las partes. A continuación se detallan dos formas de autenticación por medio de terceros.

Certificados

En la primera forma, existe un tercero llamado “Autoridad Certificante”, que emite certificados. Un certificado no es más que la clave pública de una entidad e información administrativa (indicando por lo menos quien es), firmada por ese tercero. La idea es que este tercero puede garantizar la identidad de los interlocutores al firmar el certificado. Esta firma impide que el certificado sea modificado (evitando que un intruso sustituya la clave pública por otra), lo que permite la interacción entre entidades que no se conocen directamente. Es perfectamente factible que una autoridad garantice que otra autoridad es confiable, lo que forma cadenas de confianza. Sin embargo, para que una certificación sea aceptada eventualmente debe alcanzar una autoridad en la cual se tiene confianza. Este mecanismo se usa fuertemente para autenticar sitios web y está especificada en el estandar x.509.

Web of Trust

Un método alternativo consiste en permitir que los distintos interlocutores garanticen la identidad de otros, método llamado “Web of Trust”, usado en el estandar OpenPGP. En este caso, cada usuario garantiza la identidad de los otros usuarios que conoce personalmente. Estas relaciones pueden verse como un grafo, en donde los nodos son los usuarios del sistema y los ejes son las garantías de identidad que emiten los usuarios. Un usuario que desee contactar a otro, puede buscar un camino en este grafo hacia el destino. Si lo encuentra entonces asume que está comunicándose con este otro usuario (ya que de haber sido un impostor no habría sido verificado por alguien de confianza en algún punto del camino). El sistema puede requerir múltiples garantías, pero el concepto básico es el mismo.

Ataques y temas varios

Porque es necesario un buen RNG

Los mecanismos criptográficos requieren usualmente números aleatorios (para generar claves, por ejemplo). Un punto usualmente ignorado es el generador de números aleatorios (RNG).

Uno puede tener los mejores sistemas criptográficos, pero sin un buen RNG, son completamente inútiles. Tomemos como ejemplo el envío de un mensaje encriptado con un algoritmo de clave pública. Para enviar un mensaje es necesario calcular una clave de sesión, encriptar el mensaje con ella y encriptar la clave de sesión con la clave pública del destinatario. Supongamos que el RNG es debil, basándose en la hora para generar números. En un día hay 86400 segundos, una cantidad insignificante hablando de espacios de clave. Un atacante que quisiera leer un mensaje podría lograrlo repitiendo el algoritmo de generación de la clave probando los 86400 posibles segundos, lo que le daría 86400 posibles claves para probar. Una de esas claves será la clave de sesión, por lo que probando la desencripción con las 86400 claves obtendrá una copia del mensaje.

One time pads y snake oil

Existe un algoritmo inquebrable llamado one time pad. El algoritmo consiste en tomar

una serie de bits aleatorios y realizar un XOR entre ellos y el mensaje a encriptar. El resultado es una serie de bits indistinguibles de unos bits al azar. A su vez, es imposible saber de que trataba el mensaje original, ya que cualquier mensaje puede ser resultado de la desencripción, según los bits usados como clave.

Este sistema, aunque inquebrable, es extremadamente impráctico. Primero es necesario una serie de bits aleatorios. También es necesario conservar en secreto una clave del mismo tamaño que el mensaje original. Estos dos problemas son más graves de lo que parecen. Obtener números aleatorios no es tan simple, y no es posible en un sistema determinístico. Si se intenta sustituir estos bits con el resultado de un generador de números pseudoaleatorios, esto provee una via de ataque (ya que basta con obtener la clave usada en el PRNG para quebrar el sistema). Por otra parte, si es posible almacenar una clave de un OTP en un lugar seguro, es perfectamente factible almacenar el mismo mensaje en ese lugar, lo que hace que no tenga sentido. Si bien OTP tiene sus usos, no es un mecanismo adecuado para gran parte de los usos comunes. Hay vendedores de software que venden programas para hacer “one time pad” prometiendo seguridad perfecta, pero que en realidad no es posible cumplir en la práctica, obteniendo realmente un sistema extremadamente inseguro. Este tipo de software se lo conoce como “Snake Oil” y se recomienda evitarlo a toda costa.