- -4pc - -4pc

Introducción a la Criptología

Sergio Talens-Oliag, InfoCentre [http://www.infocentre.gva.es/]
<stalens@infocentre.gva.es>

Tabla de contenidos

Resumen

En este documento daremos una breve introducción a los campos de la criptografía y el criptoanálisis, presentando su terminología, las herramientas disponibles y sus aplicaciones.

Introducción

Comenzaremos el estudio de la criptología dando algunas definiciones básicas, a continuación describiremos los sistemas criptográficos mencionando sus características más importantes, luego describiremos los distintos tipos de algoritmos criptográficos y terminaremos mencionando algunas de las aplicaciones de la criptografía.

Definiciones

* 0.60+1em

* 0.60+1em Criptología

Es el estudio de la *criptografía* y el *criptoanálisis*.

- -4pc - -4pc
* 0.60+1em Criptografía

Vista en términos sociales, es la ciencia de hacer que el coste de adquirir o alterar información de modo impropio sea mayor que el posible valor obtenido al hacerlo.

Vista en términos más formales, es la práctica y el estudio de técnicas de *cifrado* y *descifrado* de información, es decir, de técnicas para codificar un mensaje haciéndolo ininteligible (*cifrado*) y recuperar el mensaje original a partir de esa versión ininteligible (*descifrado*).

* 0.60+1em Algoritmo criptográfico

Es un método matemático que se emplea para *cifrar* y *descifrar* un mensaje. Generalmente funciona empleando una o más *claves* (números o cadenas de caracteres) como parámetros del algoritmo, de modo que sean necesarias para recuperar el mensaje a partir de la versión cifrada.

El mensaje antes de cifrar se denomina *texto en claro* y una vez cifrado se denomina *texto cifrado*.

* 0.60+1em Sistema criptográfico

Es un sistema para *cifrar* y *descifrar* información compuesto por un conjunto de *algoritmos criptográficos*, *claves* y, posiblemente, varios *textos en claro* con sus correspondientes versiones en *texto cifrado*.

Los sistemas criptográficos actuales se basan en tres tipos de algoritmos criptográficos: de clave secreta o simétricos, de clave pública o asimétricos y de resumen de mensajes (funciones de dispersión).

* 0.60+1em Algoritmos de resumen de mensajes

Transforman mensajes de tamaño variable a textos cifrados de tamaño fijo sin emplear claves. Se emplean para convertir mensajes grandes en representaciones más manejables.

* 0.60+1em Algoritmos de clave secreta o simétricos

Convierten un mensaje en un texto cifrado del mismo tamaño que el original. Emplean una sola clave para cifrar y descifrar. Son los algoritmos empleados para transferir grandes cantidades de información de modo seguro.

* 0.60+1em Algoritmos de clave pública o asimétricos

Encriptan un mensaje generando un texto cifrado del mismo tamaño que el original. Usan una clave para cifrar el mensaje (clave privada) y otra para descifrar (clave pública). Tienen un coste computacional alto y se suelen emplear para distribuir las claves de los algoritmos simétricos.

- -4pc - -4pc
* 0.60+1em Criptoanálisis

Es el conjunto de procedimientos, procesos y métodos empleados para romper un *algoritmo criptográfico*, *descifrar* un *texto cifrado* o descubrir las *claves* empleadas para generarlo.

Algoritmos de cifrado simétrico

Dentro de estos algoritmos distinguimos dos tipos de algoritmos en función de la cantidad de datos de entrada que manejan a la vez: algoritmos de cifrado por bloques y algoritmos de cifrado de flujo.

Cifrado por bloques

Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en claro y producen un bloque de tamaño fijo de texto cifrado, generalmente del mismo tamaño que la entrada. El tamaño del bloque debe ser lo suficientemente grande como para evitar *ataques de texto cifrado*. La asignación de bloques de entrada a bloques de salida debe ser uno a uno para hacer el proceso reversible y parecer aleatoria.

Para la asignación de bloques los algoritmos de cifrado simétrico realizan *sustituciones* y *permutaciones* en el texto en claro hasta obtener el texto cifrado.

La *sustitución* es el reemplazo de un valor de entrada por otro de los posibles valores de salida, en general, si usamos un tamaño de bloque k, el bloque de entrada puede ser sustituido por cualquiera de los 2^k bloques posibles.

La *permutación* es un tipo especial de sustitución en el que los bits de un bloque de entrada son reordenados para producir el bloque cifrado, de este modo se preservan las estadísticas del bloque de entrada (el número de unos y ceros).

Los algoritmos de cifrado por bloques *iterativos* funcionan aplicando en sucesivas rotaciones una transformación (*función de rotación*) a un bloque de texto en claro. La misma función es aplicada a los datos usando una subclave obtenida de la clave secreta proporcionada por el usuario. El número de rotaciones en un algoritmo de cifrado por bloques iterativo depende del nivel de seguridad deseado.

Un tipo especial de algoritmos de cifrado por bloques *iterativos* son los denominados *algoritmos de cifrado de Feistel*. En estos algoritmos el texto cifrado se obtiene del texto en claro aplicando repetidamente la misma transformación o función de rotación. El funcionamiento es como sigue: el texto a cifrar se divide en dos mitades, la función de rotación se aplica a una mitad usando una subclave y la salida de la función se emplea para hacer una o-exclusiva con la otra mitad, entonces se intercambian las mitades y se repite la misma operación hasta la última rotación, en la que no hay intercambio. Una característica interesante de estos algoritmos es que la cifrado y descifrado son idénticas estructuralmente, aunque las subclaves empleadas en la cifrado se toman en orden inverso en la descifrado.

Para aplicar un algoritmo por bloques es necesario descomponer el texto de entrada en bloques de tamaño fijo. Esto se puede hacer de varias maneras:

1. *ECB (Electronic Code Book)*. Se parte el mensaje en bloques de *k* bits, rellenando el ultimo si es necesario y se encripta cada bloque. para descifrar se trocea el texto cifrado en bloques de *k* bits y se desencripta cada bloque. Este sistema es vulnerable a ataques ya que dos bloques idénticos de la entrada generan el mismo bloque de salida. En la práctica no se utiliza.

- -4pc - -4pc

2. *CBC (Cipher Block Chaining)*. Este método soluciona el problema del ECB haciendo una o-exclusiva de cada bloque de texto en claro con el bloque anterior cifrado antes de cifrar. para el primer bloque se usa un *vector de inicialización*. Este es uno de los esquemas más empleados en la práctica.

3. *OFB (Output Feedback Mode)*. Este sistema emplea la *clave de la sesión* para crear un bloque pseudoaleatorio grande (*pad*) que se aplica en o-exclusiva al texto en claro para generar el texto cifrado. Este método tiene la ventaja de que el *pad* puede ser generado independientemente del texto en claro, lo que incrementa la velocidad de cifrado y descifrado.

4. *CFB (Cipher Feedback Mode)*. Variante del método anterior para mensajes muy largos.

Cifrado de flujo de datos

Generalmente operan sobre 1 bit (o sobre bytes o palabras de 16 ó 32 bits) de los datos de entrada cada vez. El algoritmo genera una secuencia (*secuencia cifrante* o *keystream* en inglés) de bits que se emplea como clave. La cifrado se realiza combinando la secuencia cifrante con el texto en claro.

El paradigma de este tipo de algoritmos es el *One Time Pad*, que funciona aplicando una XOR (o-exclusiva) a cada bit de la entrada junto con otro generado aleatoriamente para obtener cada bit de la salida. La secuencia de bits aleatorios es la clave de la sesión, secuencia de cifrado o el *pad*, que es del mismo tamaño que la entrada y la salida. para recuperar el texto original el texto cifrado debe pasar por el mismo proceso empleado para cifrar usando el mismo *pad*. Este algoritmo es conocido por ser el único incondicionalmente seguro, aunque, como las claves son del mismo tamaño que la entrada, es de poca utilidad práctica.

Los algoritmos de este tipo son intentos de conseguir algoritmos prácticos que se aproximen al funcionamiento del *one time pad*.

Algoritmos de clave pública o asimétricos

La criptografía de clave pública fue inventada en 1975 por *Whitfield Diffie* y *Matin Hellman*. Se basa en emplear un par de claves distintas, una *pública* y otra *privada*. La idea fundamental es que las claves están ligadas matemáticamente pero es computacionalmente imposible obtener una a partir de la otra.

Fundamentos matemáticos

Las *funciones de una sola dirección* son aquellas en las que obtener el resultado en una dirección es fácil, pero en la otra es casi imposible. Los algoritmos criptográficos de clave pública se basan en *funciones de una sola dirección con puerta trasera*, que son aquellos en los que el problema es resoluble en la dirección opuesta (la que antes era muy difícil) empleando una ayuda (la *puerta trasera*).

Los siguientes problemas matemáticos son considerados como *funciones de una sola dirección con puerta trasera* y son la base de la mayoría de algoritmos de clave pública actuales:

- *Factorización de enteros*. Un número entero siempre se puede representar como un producto de números primos denominados *factores primos*. La factorización de enteros consiste en encontrar los factores primos de un número. Los algoritmos criptográficos basados en este problema aprovechan el hecho de que la multiplicación de números primos grandes es computacionalmente sencilla pero la factorización un número grande en sus factores primos es muy cara computacionalmente.

- -4pc - -4pc

- *Logaritmos discretos*. En *aritmética módulo n* dos enteros son equivalentes si tienen el mismo resto cuando son divididos por `n`. El *resto* de la división `m/n` es el menor entero no negativo que difiere de `m` por un múltiplo de `n`. La *exponenciación discreta* ($a^x \bmod n$) es la exponenciación en *aritmética módulo n*. Por ejemplo, $3^4 \bmod 10$ es $81 \bmod 10$, que es equivalente a $1 \bmod 10$. El *logaritmo discreto* es la operación inversa a la *exponenciación discreta*, el problema es encontrar la x tal que $a^x = b \bmod n$. Por ejemplo, si $11^x = 1 \bmod 10$, entonces $x = 2$. Los algoritmos que emplean este tipo de problema se basan en que la *exponenciación módulo n* es un problema fácil y hallar el *logaritmo discreto* es un problema difícil.

- *Logaritmos discretos de curva elíptica*. Es una variación del problema anterior más cara computacionalmente, lo que permite usar claves más pequeñas que mejoran las prestaciones de los algoritmos y reducen el tamaño de los textos cifrados.

Tipos de algoritmo

En función de su relación matemática distinguimos varios tipos de algoritmo:

- *Reversible*. Es aquel en el que un mensaje cifrado con la clave privada puede ser descifrado usando la clave pública y viceversa (uno cifrado usando la clave pública puede ser descifrado usando la privada).

- *Irreversible*. Es aquel en el que un mensaje cifrado usando la clave privada puede ser descifrado con la clave pública pero la clave privada no desencripta los mensajes cifrados usando la clave pública.

- *De intercambio de claves*. Sólo permiten negociar de forma segura una clave secreta entre dos partes. Hay que indicar que los algoritmos *reversibles* también se pueden emplear para esta función, pero los *irreversibles* no.

Aplicaciones

Este tipo de algoritmos tienen dos aplicaciones fundamentales:

1. *cifrado*. Si un usuario A quiere mandar un mensaje a otro usuario B, lo encripta usando la clave pública de B. Cuando B lo recibe lo desencripta usando su clave privada. Si alguien intercepta el mensaje no puede descifrarlo, ya que no conoce la clave privada de B (de hecho, ni tan siquiera A es capaz de descifrar el mensaje).

2. *Firmas digitales*. Si B encripta un mensaje usando su clave privada cualquiera que tenga su clave pública podrá obtener el texto en claro correspondiente; si alguien quiere hacerse pasar por B tendrá que cifrar el mensaje usando la misma *clave privada* o no se descifrará correctamente con la *clave pública* de B. Lo que B ha hecho es *firmar digitalmente* el mensaje. El proceso de descifrar con una clave pública un mensaje firmado se denomina *verificación de firma*.

- -4pc - -4pc

Estos algoritmos son mucho más caros que los de clave secreta, por lo que no se usan para cifrar mucha información. Su principal aplicación está en la fase inicial de una comunicación, ya que permiten que los dos extremos se autentifiquen e intercambien claves secretas para cifrar con un algoritmo simétrico.

El problema fundamental de este tipo de algoritmos es la distribución de las claves; aunque la clave pública se puede distribuir libremente (A la puede enviar por correo o decírsela a B por teléfono), nos queda el problema de la suplantación (C le puede dar su clave pública a B haciéndose pasar por A). para solventar estos problemas se emplean autoridades certificadoras y certificados digitales, que discutiremos más adelante.

Algoritmos de resumen de mensajes

Un algoritmo de *resumen de mensajes* o *función de dispersión criptográfica* es aquel que toma como entrada un mensaje de longitud variable y produce un resumen de longitud fija. En inglés el resumen se llama *message digest*, *digest* o *hash* y el algoritmo *message digest algorithm* o *one way hash algorithm*.

Estos algoritmos deben tener tres propiedades para ser criptográficamente seguros:

1. No debe ser posible averiguar el mensaje de entrada basándose sólo en su resumen, es decir, el algoritmo es una función irreversible de una sola dirección.
2. Dado un resumen debe ser imposible encontrar un mensaje que lo genere.
3. Debe ser computacionalmente imposible encontrar dos mensajes que generen el mismo resumen.

Los algoritmos de este tipo se emplean en la generación de *códigos de autentificación de mensajes* y en las *firmas digitales*.

Códigos de autentificación de mensajes y firmas digitales

Un *código de autentificación de mensaje* (*message authentication code* o *MAC*) es un bloque de datos de tamaño fijo que se envía con un mensaje para averiguar su origen e integridad. Son muy útiles para proporcionar autentificación e integridad sin confidencialidad. para generar MACs se pueden usar algoritmos de clave secreta, de clave pública y algoritmos de resumen de mensajes.

Un tipo de MAC muy empleado en la actualidad es el *código de autentificación de mensaje resumido* (*hashed message authentication code* o *HMAC*). Lo que hacemos es generar el MAC aplicando una función de dispersión criptográfica a un conjunto formado por un mensaje y un código secreto. Así, el que recibe el mensaje puede calcular su propio MAC con el mensaje y el código secreto (que comparte con el que ha generado el MAC). Si no coinciden sabemos que el mensaje ha sido manipulado. Este tipo de técnicas se emplean para proteger comunicaciones a nivel de la capa de red.

La *firma digital* es un item que responde del origen e integridad de un mensaje. El que escribe un mensaje lo firma usando una *clave de firmado* y manda el mensaje y la firma digital. El destinatario usa una *clave de verificación* para comprobar el origen del mensaje y que no ha sido modificado durante el tránsito.

Para firmar los mensajes se emplean algoritmos de clave pública y funciones de dispersión. El proceso es como sigue:

- -4pc - -4pc

1. El emisor genera un resumen del mensaje, lo encripta con su clave privada (*clave de firmado*) y envía el mensaje y el texto cifrado que corresponde al resumen del mensaje.

2. El destinatario genera un resumen del mensaje que recibe y desencripta el resumen cifrado que lo acompañaba usando la clave pública del emisor (*clave de verificación*).>Si al comparar los resúmenes ambos son iguales el mensaje es válido y ha sido firmado por el emisor real, ya que de otro modo no se hubiera podido descifrar correctamente con su clave pública.

Hay que indicar que los MAC y las firmas digitales se diferencian en un punto importante: aunque los MAC se pueden usar para verificar la autenticidad de los mensajes, no se pueden usar para firmar los mensajes, ya que sólo se usa una clave secreta que comparten el emisor y el receptor, lo que hace que ambos puedan generar la misma firma.

Seguridad de los sistemas criptográficos

La seguridad de un sistema criptográfico depende generalmente de que al menos una de las claves empleadas sea secreta, más que de que el algoritmo de cifrado sea secreto.

El publicar los algoritmos empleados por un sistema criptográfico para que sean revisados públicamente es una buena práctica que permite que se mejoren algoritmos no totalmente seguros o se considere que un algoritmo no tiene debilidades.

Los algoritmos criptográficos tienen distintos grados de seguridad:

1. *Seguro computacionalmente*. Con suficiente poder de cálculo y almacenamiento el sistema puede ser roto, pero a un coste tan elevado que no es práctico. De cualquier modo, el coste computacional para considerar que un algoritmo es seguro ha ido cambiando con el paso del tiempo; algoritmos antes considerados seguros, como el DES, han sido rotos en meses con sistemas distribuidos y en días con sistemas diseñados específicamente para la tarea, como se describe en http://www.eff.org/descracker/.

2. *Seguro incondicionalmente*. Son aquellos en los que aun disponiendo de recursos y gran cantidad de texto cifrado no es posible romper el algoritmo. Los únicos sistemas incondicionalmente seguros son los *One Time Pads*.

Un sistema criptográfico puede ser roto en varios niveles:

1. *Deducción de información*. Se obtiene parte de información de la clave o del texto en claro.

2. *Deducción de una instancia*. Se obtiene el texto en claro a partir de un texto cifrado.

3. *Deducción global*. A partir de la deducción de una instancia se obtiene un algoritmo que obtiene los mismos resultados que el algoritmo original.

4. *Rotura total*. Se recupera la clave y se puede descifrar cualquier mensaje cifrado con la misma clave.

- -4pc - -4pc

Para romper un algoritmo se pueden emplear distintos tipos de ataque criptoanalítico:

1. *Ataque de sólo texto cifrado*. El analista dispone de un texto cifrado y quiere obtener el texto en claro o la clave. Se pueden usar métodos de *fuerza bruta* (probando todas las claves posibles hasta que obtenemos un mensaje con sentido) o basados en *diccionario* (probando únicamente con un subconjunto de las claves posibles, por ejemplo si las claves son palabras). Es importante disponer de suficiente texto en clave para que sea fácil identificar cual es el texto en claro correcto.

2. *Ataque de texto en claro conocido*. El analista dispone de un texto en claro y su correspondiente texto cifrado, lo que permite reducir el espacio de búsqueda de claves u obtener estadísticas que pueden usarse para hacer deducciones en otros textos cifrados.

3. *Ataque de texto en claro conocido adaptativo*. Es igual que el anterior pero el analista puede elegir nuevos textos dinámicamente y alterar sus elecciones en función de los resultados que va obteniendo.

4. *Ataque de texto en claro elegido*. El analista puede elegir el texto en claro y obtener el texto cifrado correspondiente. Este tipo de ataque puede evitar duplicados y centrarse más en las debilidades del algoritmo.

5. *Ataque de texto en claro elegido adaptativo*. Es la versión adaptativa del ataque anterior.

Para que un sistema criptográfico sea considerado como fuerte debe tener las siguientes características:

- Debe disponer de un número muy elevado de claves posibles, de modo que sea poco razonable intentar descifrar un mensaje por el método de la fuerza bruta (probando todas las claves).

- Debe producir texto cifrado que parezca aleatorio a un test estadístico estándar.

- Debe resistir todos los métodos conocidos de romper los códigos, es decir, debe ser resistente al criptoanálisis.

- -4pc - -4pc

Ataques más importantes sobre algoritmos criptográficos

En este apartado mencionaremos los ataques más importantes contra los algoritmos criptográficos clasificados por tipo de algoritmo.

Algoritmos de cifrado simétrico por bloques

Criptoanálisis diferencial. Se realizan sobre algoritmos de cifrado por bloques iterativos. Es un ataque de texto claro elegido que se basa en el análisis de la evolución de las diferencias de dos textos en claro relacionados cuando son cifrados con la misma clave. Mediante el análisis de los datos disponibles se pueden asignar probabilidades a cada una de las claves posibles . Eventualmente, la clave más probable puede ser identificada como la correcta.

Criptoanálisis lineal. Este es un ataque de texto en claro conocido que usa una aproximación lineal para describir el funcionamiento del algoritmo. Dados suficientes pares de texto en claro y cifrado se pueden obtener datos sobre la clave.

Explotación de claves débiles. Hay algoritmos para los que se pueden encontrar claves que se comportan de modo especial, por ejemplo dando origen a ciertas regularidades en el cifrado o un bajo nivel de cifrado. Si el número de claves débiles es pequeño no tiene importancia, pero si el algoritmo tiene muchas de estas claves es fácil que se vea comprometido.

Ataques algebraicos. Son una clase de técnicas que basan su éxito en que los algoritmos criptográficos muestren un alto grado de estructura matemática. Por ejemplo, si un algoritmo tiene estructura de grupo, al cifrar con una clave, y luego volver a cifrar con otra obtenemos un texto cifrado que podría haber sido generado con el mismo algoritmo y una sola clave, lo que hace al algoritmo bastante débil.

Algoritmos de cifrado simétrico de flujo de datos

Los principales ataques a este tipo de algoritmos buscan debilidades en la estructura del mismo que le permitan descubrir partes de la secuencia de cifrado. Una de las características fundamentales es el periodo de la clave de cifrado, ya que si es muy corto y se descubre una parte de la clave se puede emplear en sucesivos periodos del algoritmo.

Complejidad lineal. Una técnica empleada para atacar estos algoritmos es el uso de un *registro de desplazamiento lineal con realimentación* (*linear feedback shift register*) para replicar parte de una secuencia. A partir de esta técnica aparece la *complejidad lineal* de una secuencia, que será el tamaño del registro que necesitemos para replicarla.

Ataques de correlación. Otros ataques intentan recuperar parte de una secuencia de cifrado ya empleada. Dentro de estos ataques hay una clase que podemos denominar *divide y vencerás* que consiste en encontrar algún fragmento característico de la secuencia de cifrado y atacarla con un método de fuerza bruta y comparar las secuencias generadas con la secuencia de cifrado real. Este método lleva a lo que se denomina *ataques de correlación* y *ataques de correlación rápidos*.

Algoritmos de resumen de mensajes

Las funciones de dispersión deben tener dos propiedades para ser útiles en criptografía: deben ser funciones de una sola dirección y no tener colisiones. El ataque por fuerza bruta consiste en seleccionar entradas del algoritmo aleatoriamente y buscar una que nos de el valor que buscamos (la función no es de una sola dirección) o un par de entradas que generen la misma salida (la función tiene colisiones).

- -4pc - -4pc
Ataque del cumpleaños. Se trata de una clase de ataques por fuerza bruta. El nombre viene de la *paradoja del cumpleaños*: la probabilidad de que dos o más personas en un grupo de 23 personas cumplan años el mismo día es superior a `1/2`.

Si una función retorna uno de `k` valores equiprobables cuando se le proporciona una entrada aleatoria, cuando le proporcionamos repetidamente valores de entrada distintos, obtendremos dos salidas iguales después de $1.2k^{1/2}$ ejecuciones. Si buscamos una colisión en una función de dispersión, por la paradoja del cumpleaños sabemos que después de probar $1.2*2^{pi/2}$ entradas tendremos alguna.

Pseudo-colisiones. Otro problema de estos algoritmos son las *pseudo-colisiones*, que son las colisiones producidas en la función de compresión empleada en el proceso iterativo de una función de dispersión. En principio que haya pseudo-colisiones no implica que el algoritmo no se seguro.

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.