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 encriptados 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 la encriptación o un bajo nivel de encriptación. 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 encriptar con una clave, y luego volver a encriptar 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 se 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).

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.