INTRODUCCION AL CRIPTOANALISIS.

Introducción y justificación.

Existe en el mercado una amplia bibliografía dedicada a la criptografía, generalmente en inglés, aunque hay que destacar la aparición de algunos excelentes textos en castellano [PRI86][MOL94][MOR94][PIN96][FUS97][MUN97][PAS98][RAM99], y múltiples grupos de trabajo, de charla, conferencias y asociaciones. Sin embargo los trabajos sobre criptoanálisis no suelen abundar, existe poca bibliografía sobre el tema, y en general dispersa o muy anticuada. La razón de esta aparente falta de interés viene dada en general por las connotaciones que conlleva el tema. Se asocia el mismo con temas tan escabrosos como espionaje, tanto militar como industrial o diplomático, y en general cualquier forma ilícita de obtención de información. Sería muy ingenuo negar estas cuestiones, sin embargo también es muy ingenuo el hecho de considerar el criptoanálisis como un mero juego de espías. Hay una razón básica para desechar esta idea, la económica. Romper un sistema es costoso en tiempo y material, se necesita en general gran cantidad de texto cifrado y su contrapartida para intentar obtener la clave, incluso disponiendo de todo lo anterior, no podemos garantizar el éxito. Existen métodos mucho más baratos y sencillos para obtener información.

Visto lo anterior ¿Cual es la razón del criptoanálisis?. La primera y más importante el poder garantizar, con una cierta seguridad, que un sistema criptográfico no presenta fallos. Un caso famoso citado en [DWI71] es el del ataque al saliente de Saint Mihiel en Septiembre de 1918. Los americanos llevaban meses preparando un ataque para destruirlo, cuando de repente se paralizaron los preparativos, no porque el objetivo hubiera perdido su valor estratégico, ni porque el enemigo se hubiera reforzado, sino simplemente porque un estudiante de criptografía recién llegado de Washington puso en conocimiento del Cuartel General que había descifrado con relativa facilidad un mensaje secreto emitido por radio interceptado en el sector americano. Si un aficionado podía descifrar con facilidad un mensaje secreto, los expertos alemanes evidentemente también y por lo tanto debían estar al tanto de los preparativos de ataque.

Otra función del criptoanálisis es el descriptado de datos sensibles utilizados por elementos fuera de la ley, bien sea delincuentes o terroristas. Podemos citar por ejemplo un caso famoso en Estados Unidos. Un sospechoso de estar envuelto en una red de pornografía infantil fue interrogado y se le encontraron una gran cantidad de documentos cifrados, la policía estaba segura de que se trataba de fotografías de menores, pero no se le pudo acusar formalmente al negarse el susodicho a facilitar la clave de descifrado[WAY93].

Por último citar dos aplicaciones, que aunque menos evidentes, no son menos importantes. El criptoanálisis puede ser un método de introducción a conceptos matemáticos de una forma “divertida” tal como se nos indica en [KOB97], también el uso de métodos criptoanalíticos ha sido de mucha utilidad en la interpretación de escrituras y el conocimiento de lenguas muertas. Podemos citar por ejemplo los casos de Champollion y Grotefend que basándose en técnicas como el análisis de frecuencias y la asunción de una palabra probable, técnicas de uso muy frecuente en criptoanálisis, les permitieron el descifrado de la piedra de Rosetta y de las inscripciones del antiguo imperio persa.

Los primeros libros escritos sobre criptografía, estaban escritos por grandes criptoanalistas, citemos sin ir más lejos a Friedman, Sacco, Bazeries, Kerckhoffs. Estos personajes sabían como proteger la información por que conocían los métodos para desprotegerla, y sus escritos eran una mezcla de métodos criptográficos y criptoanalíticos.

Cuando la criptografía pasa de ser un "arte" a una ciencia, basado principalmente en los trabajos de Shannon, el relevo pasa de los expertos criptoanalistas a los matemáticos, siendo esta tendencia más acusada con la aparición del trabajo de Diffie y Hellman "New directions in cryptography" en el que sientan las bases de la criptografía de clave pública y proponen la utilización de funciones trampa. Se empiezan a utilizar problemas de tipo NP para producir sistemas de cifrado seguros. Se confía más en la dificultad de los problemas en sí, considerándose como válidos los métodos de cifrado basados en este tipo de problemas. Sin embargo criptoanalistas históricos como Louis Kruh y Cipher A. Deavours dudan de la validez de esta prueba al asegurar que no existe ningún teorema que nos garantice que un problema es uniformemente complejo, y que si esto no es así, y se puede encontrar un punto débil en el sistema, ese es el que utilizará el criptoanalista. Igualmente opinaba Carmona en el siglo pasado [CAR94] al definir los valores matemático y el valor material.

> *Valor* de un método es el conjunto de dificultades que éste presenta para un descifrador que desconozca la clave. Se divide en *matemático* y *material*.
>
> *Valor matemático* es el número total obtenido por el cálculo de combinaciones ó claves que un método permite emplear, y entre las cuales puede escogerse una ó varias; y por *valor material* se entiende las dificultades que independientemente de las anteriores, presenta un criptograma por la disposición o colocación de sus signos.
>
> ……...…….....……
>
> Es claro que el método que presente el mayor número de combinaciones tendrá un valor matemático mayor; pero es muy posible que no llegue a tener un *valor absoluto* igual; es decir, que no reúna tantas dificultades para el desciframiento que otro menor, matemáticamente considerado, según tendremos ocasión de observar.

El tiempo les ha dado la razón, al menos en algún caso, como el del método de Merkle-Hellman basado en el problema de la mochila. Bauer [BAU82] asimismo apunta que las medidas de complejidad utilizadas para determinar si un sistema es suficientemente seguro se basan en las cotas superiores de la misma, no estableciéndose los límites inferiores para romper el sistema.

El criptoanalista debe necesariamente estar dotado de varias virtudes, paciencia, perseverancia, imaginación, suficientes conocimientos, y sobre todo mucha suerte. La mayoría de los sistemas actuales se consideran muy seguros, y la única manera de poder romperlos es mediante un error o una mala gestión del usuario. En palabras del coronel Parker Hitt [BAU82].

> El éxito cuando se trabaja con cifrados desconocidos se mide por estas cuatro cosas en el orden nombrado: perseverancia, cuidadosos métodos de análisis, intuición, suerte.

Por otra parte debemos hacer notar que no existe un método, ni una combinación de ellos, que nos permita garantizar el resultado en un ataque criptoanalítico. Podemos hacer un símil con la medicina, no existe una cura que sirva para todas las enfermedades, sin embargo se emplean métodos de curación para cada una. Cada método se ha utilizado contra un tipo de sistema criptográfico o una combinación de ellos, siendo totalmente inoperante para otro conjunto de sistemas.

Uno de los principales problemas asociados con la asunción teórica de que el sistema es inviolable técnicamente es el descuido con el que se procede al cifrado de la información. Métodos utilizados durante siglos para evitar ataques como la sustitución homofónica, la eliminación de espacios entre palabras y otros sistemas para complicar la vida al posible

criptoanalista se obvian basándose en una teórica superioridad técnica de los métodos de cifrado, permitiendo ataques por palabra probables y la utilización de diccionarios[BAU82]. Un ejemplo es el ataque al protocolo IP mediante la utilización de palabras probables en [BEL97].

En este apartado pretendemos introducir los conceptos básicos del criptoanálisis de textos cifrados, así como los métodos más adecuados en función del algoritmo utilizado. No pretende ser exhaustivo ni excesivamente teórico, pero si dar un repaso a los métodos clásicos y una idea sucinta de los actuales.

Definición del criptoanálisis.

William F. Friedman uno de los más grandes criptólogos de la historia, famoso entre otras cosas por haber inventado el Indice de Coincidencia, definía en su libro "Elements of cryptanalysis".

> **Criptanalítica**. Criptanalítica es el nombre recientemente aplicado a la ciencia que incluye todos los principios, métodos y medios empleados en el análisis de criptogramas, es decir, su reducción o solución sin el conocimiento del sistema o la clave, o la posesión del libro de códigos, mediante un detallado estudio de los criptogramas en sí. *Criptoanálisis* es el nombre aplicado a los pasos ejecutados en la aplicación de los principios de la criptoanalítica a los criptogramas. Un *criptoanalista* es un experto en la aplicación de las operaciones o procesos en el criptoanálisis.

Actualmente la palabra criptanalítica no se usa, utilizándose la palabra criptoanálisis tanto para definir la ciencia como los pasos a seguir según la definición anterior. Además hay que recalcar que si bien el objetivo teórico sigue siendo el mismo, el conseguir descifrar el criptograma, la función del criptoanálisis actualmente suele ser más modesta. Los sistemas modernos son mucho más fuertes y están basados en sólidos fundamentos teóricos. Al entrar en el juego la comunidad científica, más preocupada por la verificación de la solidez de los principios que sustentan a los algoritmos de cifrado, el criptoanálisis tiende más a la búsqueda de posibles *debilidades* del método. Nadie considera como un ataque serio al DES el criptoanálisis lineal, ni el diferencial, son sin embargo métodos teóricos que permiten debilitar la fortaleza del DES y que en cualquier caso han mostrado debilidades muy serias en algoritmos teóricamente muy potentes como el FEAL y AKELARRE. De todas maneras el DES a pesar de las múltiples criticas que ha tenido durante su relativamente larga existencia, ha caído por pura obsolescencia y mediante ataques de fuerza bruta, no habiéndosele conseguido romper por métodos más sutiles. En [SCH98] se hace un estudio muy completo sobre la panorámica actual de la criptografía y se señalan como puntos más importantes de la obsolescencia rápida de algoritmos criptográficos el progresivo aumento de la potencia de calculo de los ordenadores, así como la aparición de una serie de técnicas que en combinación con el aumento de potencia de calculo hacen más viable el ataque a diversos algoritmos. Sin embargo el siguiente párrafo, extraído del articulo antes citado plantea la relativización de la fortaleza de los algoritmos criptográficos y hace una advertencia seria contra la fe desmesurada en las propiedades de los mismos.

> No es preciso probar todas las claves posibles, ni siquiera esforzarse por descubrir defectos en los algoritmos. Explotar los errores de diseño, implementación e instalación, representa una mejor opción. Y en la mayoría de los casos, deben aprovecharse las mismas viejas equivocaciones que los implementadores repiten una y otra vez.

> La moraleja es ente caso no es que la criptografía se inútil, sino que la criptografía no basta por sí sola. Una criptografía sólida no es la panacea. Centrarse en los algoritmos criptográficos mientras se ignoran otros aspectos de la seguridad es como intentar proteger una casa no construyendo una valla en torno a ella, sino plantando una enorme estacas en medio del jardín y esperando que el "caco" choque contra ella.

Si bien la búsqueda de debilidades puede ser, y de hecho es, una fuente de inspiración y de verificación en muchos casos de los sistemas analizados, puede llevar a errores y malas interpretaciones de resultados. Un ejemplo de esto es el criptoanálisis de los métodos mal conocidos como de la mochila. Los métodos de la mochila se basan en el problema de la suma de los subconjuntos, siendo el primero de ellos el de Merkle-Hellman. Este método fue roto por Shamir, Lagarias y Odlyzko, y Brickell. Sin embargo hay que hacer notar que solo se ha conseguido romper la implementación de Merkle-Hellman para clave pública, no habiéndose conseguido solucionar el problema en que se basa que sigue siendo NP-completo y existiendo métodos basados en él que no han conseguido romperse, como el de Chor-Rivest. Sin embargo, muchos especialistas cuando oyen hablar de criptosistemas de mochila lo asocian a un sistema débil y vulnerable. Nada más lejos de la realidad, sin embargo la aparición de una debilidad en un sistema de mochila, parece haber condenado al resto de sistemas de este tipo.

El ordenador y el criptoanálisis.

La aparición en la segunda guerra mundial de las máquinas de cifrar a rotor acaba con la época clásica del criptoanálisis. La aparición de la máquina Enigma alemana supuso un duro golpe para las secciones de criptoanálisis francesa e inglesa, que consideraron el sistema como irrompible. Sin embargo, los polacos, amenazados por los alemanes, crearon un gabinete especial de criptoanálisis dentro del Biuro Szyfrów dedicado a Enigma. En este gabinete había tres matemáticos, con dominio del idioma alemán, y entre ellos destacaba Marian Rejewski. Fue suya la idea de crear un dispositivo mecánico, conocido vulgarmente como bomba por el tictac que hacían mientras iban probando posiciones. Estos dispositivos fueron la base de las bombas más sofisticadas diseñadas por Turing. Este fue el inicio de la utilización de las máquinas en el criptoanálisis. Posteriormente, en 1943, Tommy Flowers, basándose en un diseño de Max Newman, un matemático de Blechtley Park, diseñó el primer ordenador de la historia, el Colossus, con 1500 válvulas electrónicas y programable. El Colossus fue utilizado para romper la cifra Lorenz, el sistema de cifrado utilizado por Hitler. Este se realizaba mediante la máquina Lorenz SZ40, una máquina que en esencia funcionaba como la Enigma, pero que era mucho más complicada. Desafortunadamente el Colossus y todos sus planos fueron destruidos una vez finalizada la guerra.

La utilización del ordenador como herramienta criptoanálitica de primer orden toma fuerza en Estados Unidos a partir de un informe de J. T. Pendergrass a la agencia de criptoanálisis de la Marina Americana, la OP20G, en 1946. Este informe, proponía la sustitución de la gran mayoría de los dispositivos y herramientas de propósito especial utilizados en la OP20G por un ordenador de propósito general como el de la escuela Moore, el ENIAC. En el informe incluía programas para el descifrado de la maquina Hagelin y de la Enigma de cuatro rotores, con estimación de tiempos para el criptoanálisis. El informe parece ser que fue muy convincente y obtuvo los fondos, un millón de dólares, para la construcción de una nueva computadora electrónica, la Atlas[PEN46].

Tipos de ataque.

En cualquier ataque criptoanalítico se pretende obtener la suficiente información como para poder deducir el texto original, bien sea por obtención de la clave, bien por ataques de tipo diccionario o bien por inferencia del texto en claro a partir de ciertas características del mismo y de su correspondiente cifrado.

Los tipos de ataque pueden ser:

a) activos. El criptoanalista inyecta información o modifica la circulante en el canal con vistas a obtener algún tipo de ventaja.

b) pasivos. El criptoanalista solo pretende la obtención de información y para ello se dedica exclusivamente a la escucha del canal de comunicaciones.

Para realizar sus funciones de análisis, el criptoanalista se encuentra en uno de los siguientes casos:

a) *Solo dispone de texto cifrado*. Este es el peor de los casos y evidentemente no predispone al éxito a no ser ante sistemas de cifrado verdaderamente frágiles.

b) *Se dispone del texto cifrado y del correspondiente en claro*. Existe un caso particular en el que no se conoce el texto en claro, pero sí palabras probables.

c) *Se puede disponer de cualquier criptograma correspondiente a un texto en claro escogido por el criptoanalista*. Esto no implica que se conozca el método, solo que puede obtenerse el texto cifrado.

d) *Se puede disponer del texto en claro a determinados criptogramas escogidos por el criptoanalista.* Este es caso particular del anterior. Aquí el criptoanalista no solo puede escoger el texto en claro, sino que además puede elegir éste en función de los textos en claro precedentes. Básicamente se diferencia del anterior en que en aquel se pretende tener un gran volumen de textos en claro, mientras que en éste se parte de un conjunto más pequeño y se va adaptando el texto en función de los resultados.

e) *Se conoce el algoritmo y se dispone de información suficiente que permite acotar el número de claves posibles.*

Los ataques a un sistema criptográfico intentan explotar una debilidad del mismo y pueden ser de dos tipos:

a) Ataques al propio sistema.

b) Ataques a la implementación del mismo.

En los primeros se pretende deducir una debilidad en el diseño del sistema, mientras que en los segundos la debilidad que se pretende obtener es de la implementación particular

del mismo. Este último tipo de ataques suele tener más éxito, podemos recordar sin ir más lejos las debilidades detectadas en las primeras implementaciones del SSL del navegador Netscape Navigator. Un ataque criptoanalítico pretende conseguir uno de los siguientes objetivos ordenados de mayor a menor importancia del resultado:

1. *Rotura completa del sistema*. El atacante puede conseguir cualquier clave con lo que puede obtener todos los mensajes en claro.
2. *Obtención global del mensaje*. El atacante puede obtener el mensaje en claro de cualquier criptograma sin conocer la clave.
3. *Obtención local del mensaje*. El atacante puede obtener el mensaje en claro de un criptograma dado sin conocer la clave.
4. *Obtención de una debilidad del sistema*. El atacante puede una debilidad en el cifrado que le permite atacar, ahora o en un futuro cercano, el sistema con probabilidades de éxito.

Métodos criptoanalíticos.

No existe un método general que nos permita atacar con total seguridad un sistema, por el contrario existen una gran variedad de métodos que se han utilizado con éxito contra algunas clases de cifrado. Algunos de ellos, como el criptoanálisis diferencial, tienen una importancia más teórica que practica. Debemos hacer notar que los métodos que conocemos son los que se han publicado, no sabemos nada sobre los métodos que conocen las agencias nacionales de seguridad, como caso más famoso citar que en el diseño del DES ya se había previsto el criptoanálisis diferencial varios años antes de que Biham y Shamir lo publicaran. A continuación veremos algunos de los métodos más importantes.

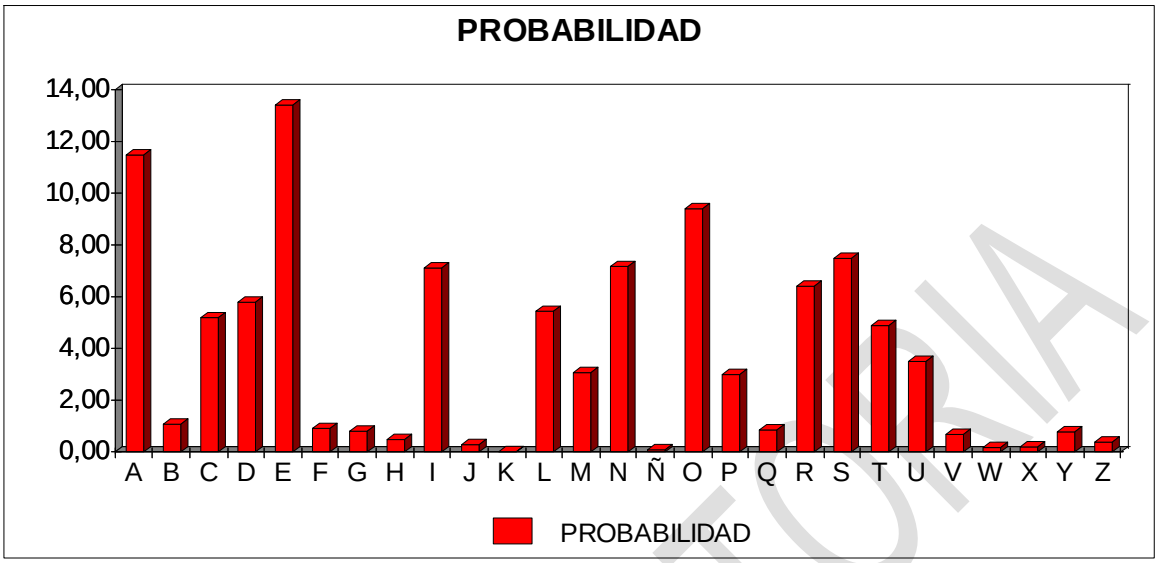
Fuerza Bruta o exhaustivo.

Es el método menos sutil, más evidente y menos practico. Este método solo es útil en los casos en los que el tamaño de la clave sea lo suficientemente reducido, ya que en promedio si existen N posibles claves deberemos examinar $\frac{N}{2}$ de ellas hasta deducir la correcta. Sin embargo la aparición de los ordenadores y en particular el espectacular aumento de potencia de los mismos ha propiciado que este tipo de ataque deba considerarse. Sin embargo debe limitarse a espacios de búsqueda limitados a no ser que se disponga de información suplementaria sobre alguna debilidad del sistema que permita reducir el espacio de búsqueda o aplicar algún tipo de búsqueda heurística.

Frecuencia de aparición de cada carácter.

Pretende utilizar las características básicas del lenguaje como son la frecuencia de aparición de los caracteres que lo forman y las combinaciones más frecuentes de ellos para reducir el espacio de búsqueda de la solución. En cualquier lenguaje existen un conjunto de letras que se repiten frecuentemente y otras que se utilizan mucho menos. También es frecuente la aparición de pares de letras (digramas) o tripletes (trigramas), todas estas características son explotadas por el criptoanalista para deducir el mensaje original a partir del

cifrado. Uno de los ejemplos más famosos de este tipo de ataque es el utilizado por el caballero Legrand en el escarabajo de oro de Edgar Allan Poe.



Método de Kasiski.

Se trata de un método de reducción de cifrados polialfabéticos inicialmente desarrollado por Charles Babbage [SIN00], pero que éste nunca publico. Quien sí lo publicó en 1860, y es por ello que se conoce con su nombre, fue el militar prusiano Friedrich W. Kasiski. Hay que indicar que el método fue desarrollado independientemente por ambos personajes.

El método se basa en comprobar la frecuencia de aparición de grupos de letras muy comunes. Si un mensaje ha sido cifrado con un alfabeto en rotación cíclica de período *P*, y una letra o grupo de letras en particular aparece *n* veces, habrá sido cifrado aproximadamente $\frac{n}{P}$ veces.

Para solucionar el criptograma, se buscan todas las repeticiones y se calcula la distancia entre ellas, el período *P* será con toda probabilidad uno de los múltiplos comunes a ellas. El resto del problema consiste en solucionar *P* cifrados de desplazamiento.

Índice de Coincidencia.

Desarrollado por W. Friedman y publicado en 1920 con el titulo de *The Index of Coincidence and Its Applications in Cryptography*, se trata de un método estadístico que nos permite entre otras cosas obtener el período en un cifrado polialfabético. Tiene su importancia por el hecho de utilizar por primera vez, de una forma sistemática, las matemáticas como una herramienta criptoanalítica. El Indice de Coincidencia *IC*, también conocido como test Kappa se basa en responder a la siguiente pregunta, dadas dos letras cualquiera de un texto ¿Cual es la probabilidad de que coincidan? Para calcularlo nos basamos en la Medida de Dispersión *MD*, que definimos como:

$$MD = \sum_{i=0}^{n-1} \left(p_i - \frac{1}{n} \right)^2$$

Sabemos, por otra parte que el número de conjuntos formados por pares de letras en un texto cifrado de longitud N es $\frac{N(N-1)}{2}$. Si denotamos por f_i la frecuencia de aparición de cada letra del alfabeto tenemos que

$$IC = \sum_{i=0}^{n} \frac{f_i(f_i - 1)}{N(N-1)}$$

Aplicando los valores de IC y MD podemos obtener el período P de un cifrado polialfabético simplemente aplicando la siguiente formula a diversos valores de P y comparando el resultado con el obtenido de aplicarlo en el criptograma.

$$\frac{(N-P).IC + (P-1).N.MD}{(N-1).P}$$

La importancia del Indice de coincidencia radica principalmente en ser la base para la posterior introducción y afianzamiento de la estadística como herramienta criptoanalítica. Es más, la estadística es la primera y más importante arma actual en el criptoanálisis. Más adelante Shannon con su teoría de la información nos dio bases teóricas más sólidas para el estudio del criptoanálisis como son el concepto de secreto perfecto, cantidad mínima de mensajes, etc. Utilizando la formula de la entropía pueden obtenerse resultados similares a los obtenidos con la aplicación del Indice de Coincidencia.

Criptoanálisis diferencial.

Desarrollado por Eli Biham y Adi Shamir, consiste en un ataque estadístico sobre texto escogido que permite deducir la clave de 56 bits del DES con 2^{47} mensajes escogidos. Básicamente pretende la obtención de información basándose en las diferencias entre valores que satisfagan unas condiciones particulares de diferencia. A medida que se van evaluando las diferencias, se asignan probabilidades a diferentes claves. La clave resultante con mayor probabilidad será la clave candidata.

El criptoanálisis diferencial funciona contra el DES y contra aquellos algoritmos que trabajen con cajas S constantes. Este tipo de ataque es muy dependiente de la estructura de las cajas S lo que obliga a establecer una estrategia de ataque diferente para cada algoritmo.

Criptoanálisis lineal.

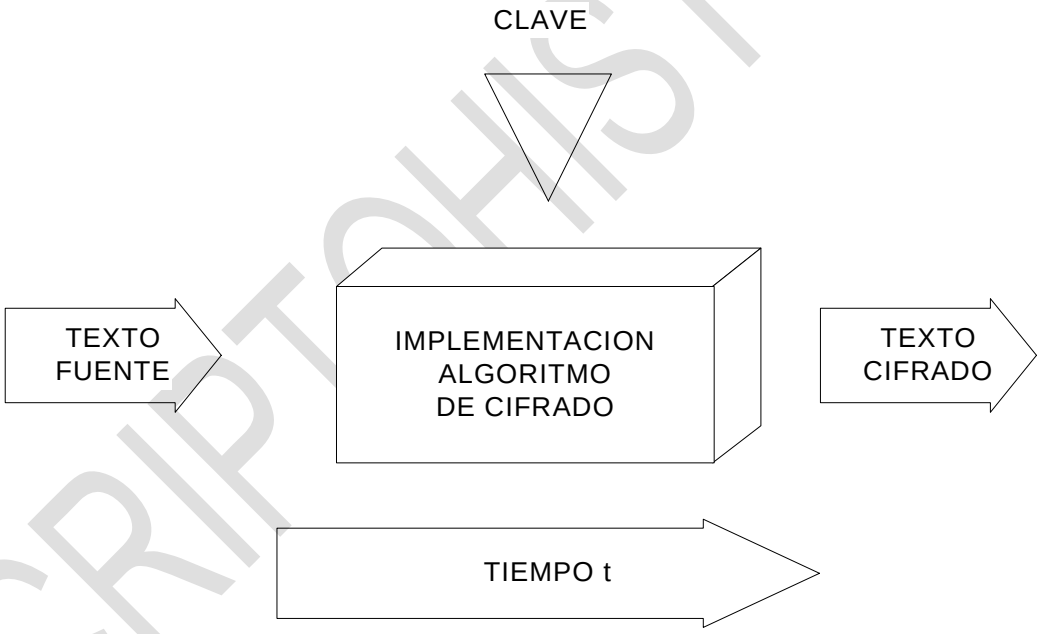
Al igual que en el caso anterior, se trata de un ataque puramente teórico, pero al contrario que el anterior, no obliga a la elección del texto a cifrar. Desarrollado por M. Matsui fue presentado en la conferencia EUROCRIPT 93. Se trata de un ataque en el cual se pretende estudiar las relaciones lineales entre los bits del texto en claro, del cifrado y las claves mediante las cuales ha sido cifrado. El método considera conocidos muchos textos en claro y sus correspondientes cifrados.

Si bien se hizo famoso como forma de ataque al DES, en realidad se trata de un tipo de ataque aplicable a cualquier sistema de cifrado en bloques que funcione con cajas *S*. En este ataque se calcula la probabilidad para todas las posibles entradas de las cajas *S* de que la paridad de entrada sea la de salida, si esta probabilidad es diferente de 0,5 existe una probabilidad de ataque. Al igual que en el caso anterior, este tipo de ataque es fuertemente dependiente de la estructura de las cajas *S*.

Existen otros ataques que pretenden la obtención de ventajas mediante la eliminación de la no linealidad de las cajas *S*. Uno de estos ataques se presenta en [FER96], que, aunque tal como reconocen los autores, no ha sido probado en la practica, es interesante como una posible nueva vía de ataque a sistemas basados en las cajas *S*.

Ataques por tiempo (Timming attacks).

Se trata de un conjunto de ataques que pretenden la obtención de información, basándose en la cantidad de tiempo necesaria para realizar las operaciones criptográficas. La base de este tipo de ataque es sencilla, es más un ataque no al sistema en sí, sino a la implementación del mismo. El esquema de cifrado básico sería el mostrado en el esquema siguiente:



Como vemos en el dibujo anterior, un texto fuente es pasado a una implementación de un cifrador que nos generará un texto cifrado en un tiempo *t*. Evidentemente, este tiempo *t* dependerá de la implementación del algoritmo, de los datos de entrada y de la longitud de la clave. Suponiendo fija la implementación del algoritmo, podemos obtener información muy importante del tiempo de calculo del resultado. Este tipo de ataque es muy adecuado en general contra sistemas de clave pública como el RSA en los cuales los tiempos de cifrado y descifrado son fuertemente dependientes del tamaño y las características de las claves. [KOC**]

Ataques basados en fallos de hardware.

Se trata de un tipo de ataques que fueron originalmente creados por la comunidad hacker para prolongar las subscripciones a las TV de pago. Son especialmente adecuados para el criptoanálisis de tarjetas chip. Básicamente el ataque pretende inducir errores en el funcionamiento del dispositivo de cifrado mediante una perturbación física tal como la aplicación de radiación de iones o cualquier tipo de perturbación que cause un mal funcionamiento del dispositivo. Se pretende que el dispositivo genere errores con diferencias de un bit. El ataque puede hacerse bien induciendo pequeños cambios en la clave, en los datos o bien en el código. Se trata de un ataque muy potente, en [BIH96] se indica que es posible recuperar una clave del DES simplemente con 200 textos cifrados, con fallos aleatorios de un bit en todos las vueltas. El ataque se ha demostrado útil en otros sistemas como el RSA [AND**][BIH96][***96][BON**].

Ataque del diccionario.

En este tipo de ataques no se pretende la obtención de la clave, sino directamente del texto en claro ya que el método de cifrado es público, y aunque no se disponga de la clave se puede reproducir. Este es el caso de los sistemas de cifrado de claves de acceso en Sistemas Operativos. Cuando un usuario quiere darse de alta en un sistema, introduce su código y su clave de acceso, ésta es cifrada y posteriormente se compara el resultado con la clave cifrada que se almacena en el fichero de claves. Si son iguales el sistema considera que el usuario es quien dice ser y le permite el acceso. Los programas rompedores de password parten del hecho de que se ha obtenido una copia del fichero de claves (el /etc/passwd de UNIX por ejemplo) y comprueban si existe alguna cuenta sin clave, en cuyo caso utilizan, y con el resto se realiza el siguiente ataque. Por una parte se dispone de un diccionario en claro con una gran cantidad de palabras y combinaciones de ellas muy comunes y validas como claves. Posteriormente se realiza el cifrado de las mismas con la utilidad del sistema a atacar, o una copia de la misma, se comprueba el resultado del cifrado con el contenido del fichero de claves y en el caso de que se produzca una coincidencia inferimos cual es la palabra en claro.

Meet in the Middle.

Se trata de un ataque del punto medio contra cifrados dobles en el que se supone conocido el texto original y su equivalente cifrado. Por una parte se cifra un texto en claro con una clave k_1 y se descifra el equivalente cifrado con otra clave k_2. Cuando ambos resultados coincidan se dispone de las dos claves. Este ataque obliga a almacenar todos los bloques de cifrado realizados con k_1 para comparación. En un caso como el DES supondría almacenar $2^{56}.64$ bits.

Una versión más sofisticada del mismo se basa en la resolución del siguiente problema. Sea un universo de *m* elementos y dos muestras aleatorias de tamaño *r*. ¿Cuál es la probabilidad de que los dos conjuntos seleccionados sean disjuntos? La probabilidad de que esto ocurra viene definida por la formula

$$p_{m,r} = \frac{((m-r)!)^2}{(m-2r)!\,m!}$$

Canales subliminales.

La utilización de software de cifrado de manera masiva, ha llevado a la siguiente paradoja, se utiliza software de cifrado para proteger la información de terceros, pero no se tiene la garantía de que el software no permita la recuperación de información por parte de terceros. Es totalmente factible el hecho de que el software que utilicemos para enviar la información cifrada por Internet, envíe en paralelo y sin nuestro consentimiento el mismo mensaje en claro a otra dirección prefijada. Este caso es relativamente fácil de comprobar, mucho más difícil es la comprobación de que nuestro software no genere información extra que permita la recuperación por parte de un tercero, lo que se denomina un *canal subliminal*. Para realizar un canal subliminal por parte del programador bastaría con que se realizasen los siguientes pasos:

1) Generar con una semilla aleatoria un par de claves pública y secreta del programa.
2) Cifrar la semilla aleatoria con la clave pública del programador.
3) Introducir la semilla cifrada en la clave pública del programa.

De esta manera se obtiene una clave pública modificada e indetectable que permitirá al programador volver a obtener la clave secreta simplemente descifrando la semilla y volviendo a generar las claves.

Ataques a protocolos.

El ataque a un sistema criptográfico requiere una gran cantidad de potencia de calculo, información, habilidad y sobre todo suerte. Es mucho más factible el ataque a los protocolos, ya que lo que se pretende con ellos es mediante un error de uno o ambos comunicantes obtener información que nos permita obtener el resultado que deseamos.

Los ataques más frecuentes suelen ser:

- *Ataque con clave conocida.* Consiste en la utilización de claves de cifrado conocidas utilizadas en anteriores transacciones.

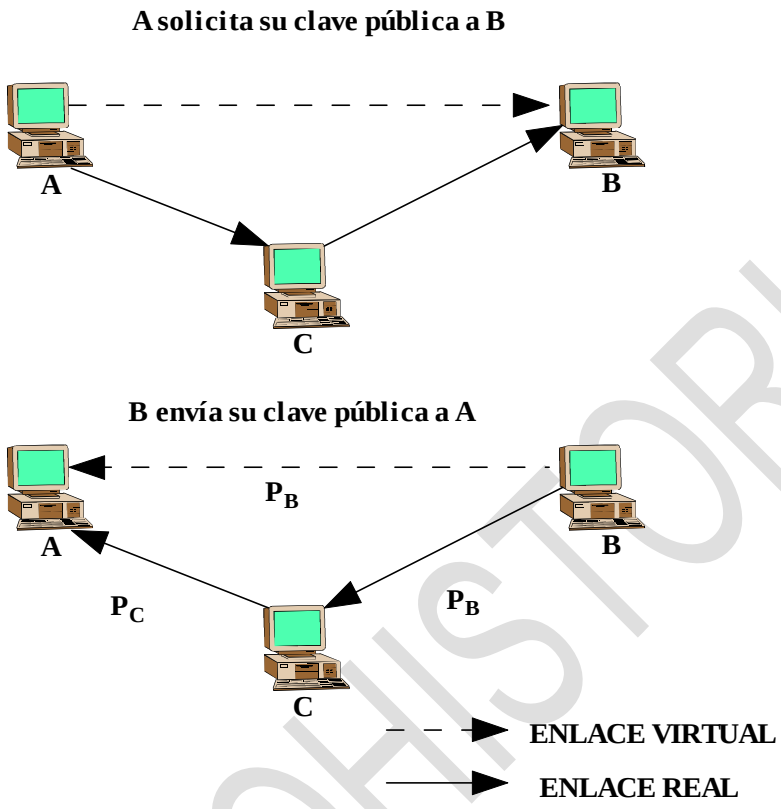
- *Repetición.* Consiste en la reutilización de mensajes anteriormente utilizados.

- *Reflexión.* Consiste en responder al autor del mensaje con el mismo mensaje.

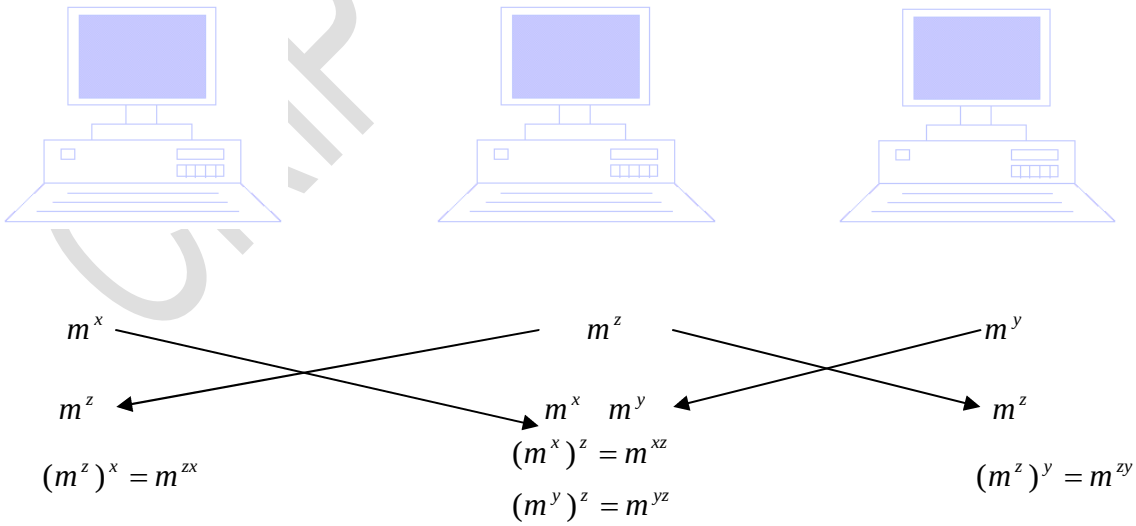
- *Suplantación.* Consiste en la suplantación de la personalidad de un usuario.

- *Ataque del intermediario(man in the middle).* Se trata de un ataque a protocolos de clave pública que puede resultar muy peligroso, aunque en sí es bastante sencillo. Consiste básicamente en una suplantación doble; el atacante se sitúa en medio de los dos usuarios legales realizando una intervención directa en el tráfico entre ambos. Para ello no hace falta decir que es necesario que el atacante consiga introducirse en la red y tomar control de las comunicaciones. En este caso supongamos que el usuario *A* necesita conectarse con el usuario *B* de una forma segura. *A* solicita a *B* su clave pública, pero el mensaje de *B* a *A* es interceptado por el usuario *C*, que en su lugar envía a *A* su propia clave pública. El

resto es sencillo, cuando *A* quiera enviar un mensaje a *B*, éste será cifrado con la clave pública de *C*, que lo recibirá, descifrará y reenviará a *B* cifrándola con su clave pública. De esta manera *C* será capaz de leer todas las comunicaciones entre *A* y *B*.



A continuación podemos ver un ejemplo, un poco más formal, utilizando el algoritmo de intercambio de claves de Diffie-Hellman.



x, y, z: números secretos.

Ataque del cumpleaños.

Se trata de un ataque basado en la paradoja del cumpleaños. Esta paradoja se presenta de la siguiente manera ¿Cual es el mínimo valor de *k* para que la probabilidad de que haya como mínimo dos personas con el mismo cumpleaños sea mayor que 0´5? Ignorando los años bisiestos y considerando todos los días equiprobables podemos calcular que con un valor de 23 tenemos una probabilidad de 0´5073. Formalmente, la probabilidad de que en un grupo de *m* elementos (días), haya dos de ellos con el mismo valor en un grupo de *n* elementos (personas) es de:

$$p_{m,n} = \frac{m!}{(m-n)!\,m^n}$$

Supongamos ahora que estamos utilizando códigos de autentificación de mensajes (MAC) de 64 bits y sea H la función de hash. Podemos suponer que un atacante que quisiera obtener un M´ tal que H(M)=H(M´) necesitaría en promedio hacer 2^{63} intentos. Basándonos en la paradoja del cumpleaños podemos utilizar una estrategia de ataque con solo 2^{32} intentos. El ataque es como sigue:

1) El origen *A* genera un MAC de *m* bits y lo cifra con su clave privada.
2) El oponente genera $2^{\frac{m}{2}}$ variaciones en el mensaje, en todas ellas el significado del mensaje es el mismo. Prepara además un número igual de mensajes, todos ellos son variaciones del mensaje fraudulento que sustituirá al correcto.
3) Se comparan los dos conjuntos de mensajes hasta encontrar un par que generen el mismo código. La probabilidad de éxito, según la paradoja del cumpleaños, es mayor que 0´5. Si no se consigue encontrar un par, se siguen generando pares hasta que haya uno que coincida.
4) El oponente ofrece la variación valida a *A* para que la firme. Esta firma puede añadirse al mensaje fraudulento ya que tienen el mismo MAC, el oponente ha conseguido su objetivo incluso sin conocer la clave de cifrado.

Análisis de tráfico.

El fin último del criptoanálisis es la obtención de información no autorizada y en la practica por cualquier medio. El análisis de tráfico estudia las variaciones en la cantidad y características de los mensajes para obtener información por inferencia. En caso de conflicto bélico y si suponemos que el enemigo está preparando una ofensiva, un buen indicador del inminente comienzo de la misma es el aumento de la cantidad de mensajes. La dirección de los mensajes puede indicarnos por ejemplo cual es el centro neurálgico o de toma de decisiones. En espionaje industrial puede indicarnos donde se está realizando un determinado proyecto o desde donde se dirige. Como norma el flujo de información por la red de comunicaciones debe mantenerse constante para evitar este tipo de ataques.

Criptoanálisis de cifradores en flujo.

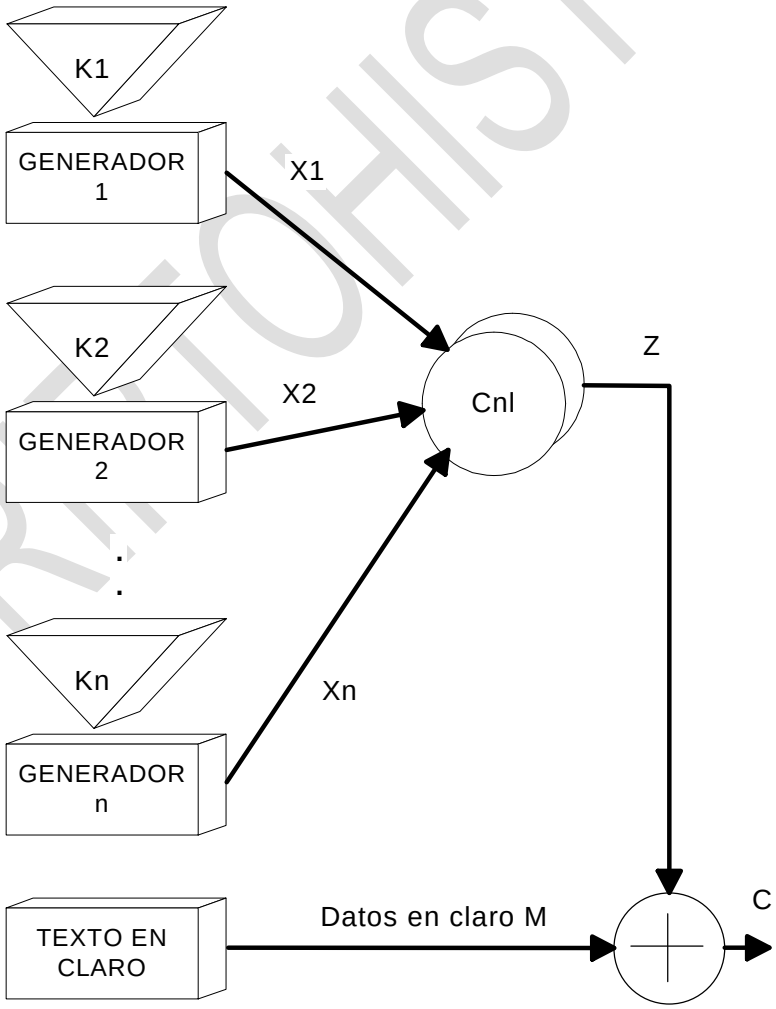
Criptoanálisis de registros de desplazamiento lineal.

Suponemos conocido un texto en claro $x_1, x_2, .., x_n$ y su correspondiente cifrado $y_1, y_2, ..., y_n$, con lo que podemos calcular los bits de la clave $z_i = x_i + y_i \bmod 2$ con $1 \le i \le n$. Supongamos conocido el valor de *m*, si $n \ge 2m$ podemos plantear un sistema de *m* ecuaciones con *m* incógnitas, que posteriormente puede ser resuelto. El sistema de *m* ecuaciones sería el siguiente:

$$(z_{m+1}, z_{m+2}, .., z_{2m}) = (c_0, c_1, .., c_{m-1}) \begin{pmatrix} z_1 & z_2 & \dots & z_m \\ z_2 & z_3 & \dots & z_{m+1} \\ z_{1m} & z_{m+1} & \dots & z_{2m-1} \end{pmatrix}$$

Ataque por correlación de Siegenthaler.

Este ataque está orientado a los generadores formados por una combinación a través de una función no lineal de varios LFSR. El modelo busca investigar la dependencia estadística introducida por el combinador no lineal C_{nl}.



Siegenthaler estableció que la salida del combinador no lineal puede verse como una perturbación sobre cada una de las secuencias $x_i(t)$, y que esta puede verse como definida por una fuente de ruido sin memoria con probabilidad *p*. Si esta probabilidad *p* es menor que $\frac{1}{2}$ puede intentar determinar las claves de cada secuencia x_i calculando la correlación de éstas con la secuencia de salida. El proceso sigue los siguientes pasos[GUI98]:

1. Realizar una búsqueda exhaustiva de las claves K_i de cada una de las $x_i(t)$.
2. Seleccionar ciertos valores de las claves K_i examinando la correlación entre la salida $z(t)$ y los distintos valores de K_i que se van probando.

Ataque por consistencia lineal.

Se trata de un test que pretende la obtención de la clave por búsqueda exhaustiva de una porción de la misma. Se basa en que en ciertos casos la tasa de redundancia de la clave $\rho < 1$, siendo $\rho = \frac{|k| - |k_1|}{|k|}$, con lo que la seguridad real del sistema depende de solo k_1 bits de la clave. El sistema se basa en la determinación de consistencia de un sistema de ecuaciones cuya matriz de coeficientes depende únicamente de k_1. Los pasos a seguir en este caso son[GUI98]:

1. Generar una matriz *A* de coeficientes aleatorios obtenida a partir de k_1 bits de la clave. Dicha matriz consta de *m* filas por *n* columnas, con $m > n$.
2. Considerar una porción de *n* bits de la salida como el vector no nulo *b*.
3. Comprobar que el sistema $A.x = b$ es compatible, en cuyo caso la clave k_1 es la correcta y solo queda deducir el resto de la clave a partir de la solución del sistema. En caso contrario hay que probar otro k_1.

Ataque por síndrome lineal.

Se trata de un método de ataque desarrollado por Zeng y Huang. Este método se utiliza cuando las secuencias generadas por la fuente de texto en claro no están estadísticamente equilibradas, es decir, contiene más ceros que unos. Este método permite romper generadores como el stop and go de Beth Piper y el de Geffen.

El proceso consiste en considerar que la secuencia binaria proviene de un generador LFSR cuyo polinomio característico *f(x)* es conocido, pero sin embargo, no es conocido su estado actual.

Si denominamos *m*(*t*) a la secuencia de texto en claro, *c*(*t*) a la secuencia de texto cifrado, y *s*(*t*) a la secuencia pseudoaleatoria, tenemos que:

$$c(t) = m(t) \oplus s(t)$$

$$p = prob(s(t) = c(t)) = \frac{1}{2} + \varepsilon \text{ con } \varepsilon > 0$$

Pretendemos obtener $s(t)$ a partir de $c(t)$ incrementando ε. Los pasos a seguir son:

1. Considerar un conjunto de múltiplos trinomiales de $f(x)$ de la forma $g(x) = 1 + x^{i_1} + x^{i_2}$ con $i_2 > i_1 > 0$.
2. Calcular para cada trinomio los tres síndromes siguientes:

$$\sigma_1 = c(t) \oplus c(t+i_1) \oplus c(t+i_2)$$
$$\sigma_2 = c(t+i_1) \oplus c(t) \oplus c(t+i_2-i_1)$$
$$\sigma_3 = c(t-i_2) \oplus c(t-i_2+i_1) \oplus c(t)$$

3. Si consideramos $2m+1$ síndromes. Construir una nueva señal $c'(t)$ de acuerdo con la siguiente regla:

$$c'(t) = \begin{cases} \overline{c}(t) & \text{si al menos m} + 1 \text{ síndromes son 1} \\ c(t) & \text{en otro caso} \end{cases}$$

Se demuestra que ε_m tiende a $\frac{1}{2}$ cuando para m muy grande, esto implica que podremos recuperar $s(t)$ y en consecuencia $m(t)$ a partir de $c(t)$.

Criptoanálisis específicos.

En este apartado vamos a ver el criptoanálisis de dos métodos de cifrado de clave pública, el de Merkle-Hellman y el del RSA, un ataque al cifrado de Hill y por último una serie de ataques contra el DES. De los dos métodos de clave pública, el primero ha sido forzado como ya hemos indicado anteriormente, el segundo sin embargo resiste por el momento a todos los ataques realizados en su contra, los ataques por tiempo y por hardware se han demostrado como válidos en algunos casos, siendo el primero fácil de evitar, bien sea generando retrasos aleatorios o simplemente asegurando que el tiempo de exponenciación sea constante. Sin embargo, hoy por hoy el principal enemigo que podría tener el RSA sería la aparición de un método rápido de factorización que podría materializarse no solo por la investigación matemática en ese campo, sino también por los vances en computación cuántica.

Criptoanálisis del método de Merkle-Hellman.

El criptoanálisis del método de Merkle-Hellman se basa en el análisis de los números que forman la mochila fuerte para encontrar un par N y M tales que el conjunto $\{Mw_i' \pmod N\}$ sea una secuencia supercreciente cuya suma es menor que N. Finalmente se aplica el algoritmo de Lenstra-Lenstra-Lovász (L^3) para resolver un conjunto de ecuaciones de programación entera. Veremos a continuación un ejemplo de una versión reducida del algoritmo obtenida de [PAT87], puede utilizarse la misma referencia para ver el criptoanálisis completo tanto de Shamir como el de Lagarias/Odlyko y el de Brickell.

Sea $\{w_i'\} = \{467,355,131,318,113,2,135,215\}$ el conjunto de números público. Debemos encontrar un par M,N tales que $\{Mw_i' \pmod N\}$ sea un conjunto supercreciente. Hagamos la hipótesis de que el orden de los números corresponde al orden creciente de los números en la

mochila fácil. Lo que significa que al ser un número como mínimo dos veces el anterior, tenemos que $M.w_i^{'} \equiv 467.M$ será menor que $2^{-7}.N$.

Si buscamos los ceros de la función $f(M)=467M(\mathrm{mod}N)$ encontraremos que son $M = \frac{N}{467}, \frac{2N}{467}, \ldots$, como además los valores de $f(M)$ deben ser menores que $2^{-7}.N$, entonces $M \in \left[\frac{pN}{467}, \frac{pN}{467} + \frac{2^{-7}N}{467}\right]$ con p =1,...,466. Dividiendo por N tenemos que $\frac{M}{N} \in \left[\frac{p}{467}, \frac{p}{467} + \frac{2^{-7}}{467}\right]$. Aplicando el mismo razonamiento al número 355 tenemos que $M.w_2^{'} \equiv 355.M$ será menor que $2^{-6}.N$. Con lo cual tenemos que $\frac{M}{N} \in \left[\frac{q}{355}, \frac{q}{355} + \frac{2^{-6}}{355}\right]$ con q =1,..,354. Ya que la información que buscamos $\frac{M}{N}$ debe caer en ambos intervalos, debemos calcular la siguiente expresión: $\left(\bigcup_{1}^{466}\left[\frac{p}{467}, \frac{p}{467} + \frac{1}{467.128}\right]\right) \cap \left(\bigcup_{1}^{354}\left[\frac{q}{355}, \frac{q}{355} + \frac{1}{355.64}\right]\right)$. Los calculos nos dan la siguiente lista de pares:

```
[0’053533190578, 0’053565140845]
[0’107066381156, 0’107086267606]
[0’160599571734, 0’160607394366]
[0’473239436620, 0’473250133833]
[0’526766595289, 0’526804577465]
[0’580299785867, 0’580325704225]
[0’633832976445, 0’633846830986]
[0’687366167024, 0’687367957746]
[0’946478873239, 0’946483538544]
```

Haciendo la misma operación con el resto de los números que forman la mochila vamos reduciendo el número de pares hasta obtener un conjunto reducido de pares que nos permitan establecer una serie de ecuaciones de la forma $\frac{q}{w_{.j}} \le \frac{p}{w_i^{'}} \le \frac{q}{w_{.j}}(1 + \frac{1}{w_j^{'}.2^k}$,en este caso la solución es posible mediante la utilización del algoritmo L^3, una descripción del mismo puede encontrarse en [PAT87] y [MEN97]. Una vez encontrado el intervalo [*x*, *y*], necesitamos encontrar una solución a la expresión $x \le \frac{M}{N} \le y$ limitado por *N*. En el ejemplo que nos ocupa tenemos como intervalos resultantes después de aplicar $w_4^{'}$ `[0’053533190578, 0’053565140845]` y `[0’107066381156, 0’107086267606]` con lo cual solo nos queda por responder que posibles valores de *M* y *N* dan una fracción cuyos valores caen en los intervalos anteriores. Es decir tenemos que encontrar valores para *M* y *N* de forma que se cunplan las siguientes restricciones con mcd(*M*,*N*) = 1.

$$N \le 2.max\left|w_i^{'}\right|$$

$$`0'053533190578 \le \frac{N}{M} \le 0'053565140845$$

$$0'107066381156 \le \frac{N}{M} \le 0'107086267606$$

los dos pares que cumplen la siguiente condición son (25,467) y (53,990), en el primer caso el resultado es: 0,2,6,11,23,58,106,238 y en el segundo 1,5,13,24,49,123,225,505. En este caso el resultado correcto es el segundo y lo que tendríamos que hacer una vez interceptado el mensaje sería calcular $53W \pmod{990}$, resolver el mensaje para los números {1,5,13,24,49,123,225,505} y obtendriamos el mensaje descifrado.

Criptoanálisis del DES.

El DES ha sido uno de los sistemas más analizados de la historia. Los intentos de romperlo han sido frecuentes y en la mayoría de los casos inútiles. Al igual que el RSA podemos concluir que se trata de un algoritmo muy bien diseñado y que ha resistido con admirable fortaleza todos los ataques que ha recibido. Sin embargo, el esfuerzo y empeño en romper este algoritmo ha hecho aparecer toda una serie de nuevos métodos de criptoanálisis, algunos de los cuales han sido fatales para otros algoritmos.

Hoy por hoy, la principal debilidad del DES sigue siendo el tamaño de su clave. Una posibilidad de "aumentar" el tamaño de la clave sería el doble cifrado, pero esto no representa ninguna ventaja ya que para el criptoanálisis podemos utilizar un ataque del tipo *meet in the middle* que nos reduce el problema al de criptoanalizar un mensaje con una sola clave. El futuro actual del DES pasa por la utilización del triple DES, es decir la aplicación del DES tres veces con dos o tres claves de cifrado que aumentaría el espacio de claves hasta 2^{112}. Lo que es indudable es que con el tamaño actual de clave (56 bits) la vida útil del DES es muy limitada, los ataques al mismo son cada vez más frecuentes y los sistemas actuales permiten la recuperación de la clave en tiempos relativamente cortos. El último ataque realizado con éxito ha conseguido reventar el algoritmo en 22 horas y 15 minutos. Esta hazaña ha sido realizada por una maquina especial de la Electronic Frontier Foundation. Para obtener más información de este logro puede consultarse en Internet la pagina www.eff.org/pub/Privacy/Crypto_misc/DESCracker/HTML/199901119_deschallenge3.html o leer el esquema y diseño completo del ataque en la referencia [EFF98].

Claves débiles.

Son todas aquellas claves que cumplen que dado cualquier mensaje el cifrado doble del mensaje con la misma clave da como resultado el mensaje original. Existen cuatro claves débiles conocidas que son:

Representación en hexadecimal incluyendo bits de paridad

0101010101010101	FEFEFEFEFEFEFEFE
E0E0E0E0F1F1F1F1	1F1F1F1F0E0E0E0E

Representación en binario sin incluir bits de paridad [WOB07, Página 157]

0000 000 0000 000 0000 000 0000 000	0000 000 0000 000 0000 000 0000 000
1111 111 1111 111 1111 111 1111 111	1111 111 1111 111 1111 111 1111 111
1111 111 1111 111 1111 111 1111 111	0000 000 0000 000 0000 000 0000 000

```
0000 000 0000 000 0000 000 0000 000   1111 111 1111 111 1111 111 1111 111
```

Existen además seis parejas de claves semidebiles, que son aquellas en las cuales el cifrado con la segunda clave sobre el cifrado de la primera genera el mensaje original. Estas son:

Representación en hexadecimal incluyendo bits de paridad

```
01FE01FE01FE01FE,FE01FE01FE01FE01
1FE01FE00EF10EF1,E01FE01FF10EF10E
01E001E001F101F1,E001E001F101F101
1FFE1FFE0EFE0EFE,FE1FFE1FFE0EFE0E
011F011F010E010E,1F011F010E010E01
E0FEE0FEF1FEF1FE,FEE0FEE0FEF1FEF1
```

Criptoanálisis diferencial.

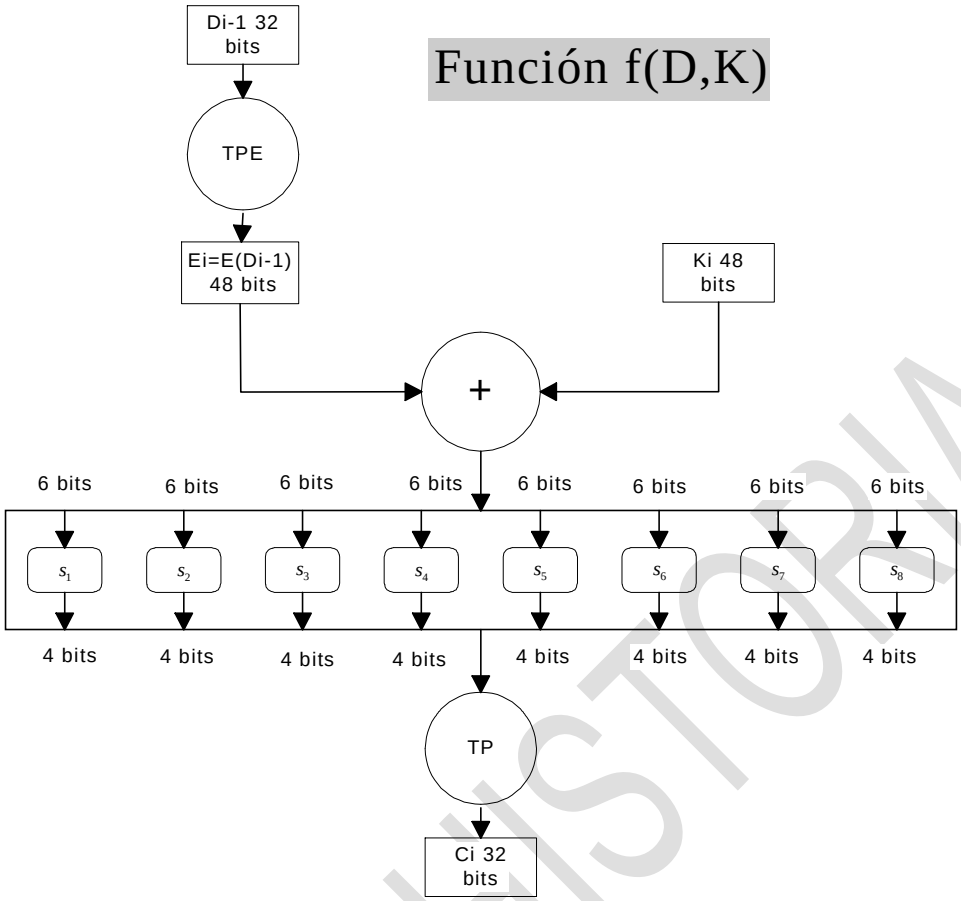
Este tipo de ataque no representa una amenaza practica a la fortaleza del DES. Se trata de un ataque más teórico que practico, ya que en circunstancias normales no se tiene el acceso al dispositivo de cifrado para realizar el cifrado de los textos escogidos. Se ha de hacer notar que los diseñadores del DES conocían este tipo de ataque e hicieron resistente al DES contra el mismo tal como reconoció Dan Coppersmith de IBM, pero que los detalles de este tipo de ataque en su día estaban clasificados por el gobierno de los Estados Unidos.

Sea S_i una de las cajas *S* del DES, K_i la subclave asociada a la caja S_i y C_i y C_i^* dos salidas de la caja obtenidas a partir de dos textos en claro escogidos. El ataque se basa en la reducción de los posibles valores de K_i obtenidos a partir de las salidas de las cajas.

La obtención de las K_i se realiza calculando las 32 posibles parejas de entrada de la caja S_i y basándose en que $C_i \oplus C_i^* = (C_i \oplus K_i) \oplus (C_i^* \oplus K_i)$ se escogen las parejas de valores *x, y* tales que $S_i(x) + S_i(y) = S_i(C_i \oplus K_i) \oplus S_i(C_i^* \oplus K_i)$ con lo que

$$K_i = \begin{cases} (C_i^* \oplus x) = (C_i \oplus y) \\ o \\ (C_i \oplus x) = (C_i^* \oplus y) \end{cases}$$

Repitiendo este proceso se puede encontrar el valor correcto de K_i [PAS98].



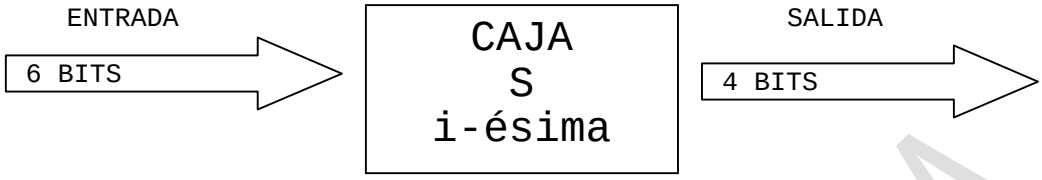
Criptoanálisis diferencial con fallos.

Este ataque fue presentado por Eli Biham y Adi Shamir[BIH97] y en esencia es una variación del criptoanálisis diferencial para aprovechar la información generada por errores introducidos en el cifrado. Se trata de un ataque diseñado principalmente para el criptoanálisis de tarjetas inteligentes. Si no se producen muchos errores y estos ocurren al final del proceso de cifrado puede inferirse información de las diferencias entre el cifrado correcto y el inducido por errores. Según los autores en una simulación se necesitaron entre 50 y 200 cifrados para determinar la clave. En dicha simulación se generaron errores en todas las vueltas, pero todos ellos eran conocidos. En la practica generar errores en posiciones determinadas y en determinadas vueltas del DES es bastante poco probable[PAS98].

Criptoanálisis lineal.

Este ataque fue propuesto por M. Matsui en 1993 y pretende la descripción del DES mediante aproximaciones lineales. Básicamente el sistema pretende la obtención de relaciones lineales en las cajas *S*, ya que el resto de las operaciones en el DES son lineales Para ello se obtienen un conjunto de ecuaciones que representen la relación existente entre ciertos bits del

mensaje en claro y otros del cifrado. Sean S_i y E_i la salida y la entrada de la i-ésima caja *S* del DES.



Existen 63 posibilidades de combinación útiles de los bits de entrada en E_i y 15 de salida de S_i se estudia la posibilidad de que una combinación de entrada produzca una de salida con una probabilidad alta. Para ello se suman con una O exclusiva todos los bits de entrada y de salida para generar un bit de la clave. Si la probabilidad difiere bastante de 0,5 en un sentido u otro se almacena la ecuación resultante. Procediendo de esta manera se van generando ecuaciones que permitirán obtener la clave. En el DES serían necesarios 2^{47} textos en claro para obtener la clave.

Criptoanálisis del RSA.

Ataque basado en la factorización de la clave pública.

Evidentemente el ataque más claro al RSA consiste en intentar factorizar *n*. La búsqueda de nuevos métodos de factorización es el principal campo de investigación en el criptoanálisis del RSA, sin embargo sigue siendo mucho más fácil determinar si un número es primo que obtener los factores de un número que se sabe que no lo es. El mayor número factorizado hasta el momento tiene 155 dígitos (512 bits) y para romperlo se necesitaron 292 ordenadores trabajando 5´2 meses[RSA99]. Pero, ¿existen otros posibles ataques al algoritmo?, el primero y más evidente es intentar obtener $\phi(n)$. Si obtenemos $\phi(n)$, y ya que *n* es producto de dos primos *p* y *q*, entonces *n* puede ser fácilmente factorizado resolviendo las siguientes ecuaciones:

$$n = p.q$$
$$\phi(n) = (p-1).(q-1)$$

sustituyendo q por $\frac{n}{p}$ obtenemos la siguiente ecuación $p^2 - (n - \phi(n) + 1)p + n = 0$, con lo cual si un criptoanalista consigue averiguar el valor de $\phi(n)$ puede factorizar *n* y romper el código. Desgraciadamente calcular $\phi(n)$ no es más fácil que factorizar *n*.

Otro posible ataque sería intentar determinar *d* sin factorizar *n*. Si fuese posible obtener *d*, podríamos calcular $e.d-1$ que es un múltiplo de $\phi(n)$. Sin embargo si *n* es lo suficientemente grande, no es más fácil obtener *d* que factorizar *n*.

Claves débiles en el RSA[LUC99].

Existen una serie de valores para los que se cumple que $M^e = M \bmod n$. En general esta posibilidad ocurre siempre independientemente de cual sea el valor de *n*. El número de valores que quedan igual al ser codificados viene dado por la formula

$$\sigma_N = (1 - mcd(e-1, p-1))(1 - mcd(e-1, q-1))$$

Ataques por proximidad.

Este tipo de ataque puede realizarse cuando se suponer que *p* y *q* son número muy próximos y se basa en el método de factorización de Fermat[COU99]. Si ambos números cumplen el estar muy próximos, y suponiendo que $p > q$ tenemos que $n = p.q$, al ser muy próximos tenemos que $\frac{p+q}{2}$ y $\frac{p-q}{2}$ son ligeramente mayor y menor que $\sqrt{n}$ con lo que $n = \frac{(p+q)^2}{4} - \frac{(p-q)^2}{4}$. Si denotamos como $x^2 = \frac{(p+q)^2}{4}$, $y^2 = \frac{(p-q)^2}{4}$, tenemos que la solución a la ecuación $n = x^2 - y^2$ viene dada cuando $x^2 - n$ es un cuadrado perfecto. En este caso, $p = x + y$, $q = x - y$. [PAS98][SAL96].

6.7.3.4 Ataque por ocultamiento.

Es un ataque que solo puede tener éxito en el caso de que uno de los comunicantes participe inadvertidamente[BON99]. Supongamos que un personaje malintencionado que llamaremos Matías presenta un mensaje *M* a Alicia para que lo firme. Este mensaje es comprometedor y Alicia se niega a firmarlo. Matías crea un nuevo mensaje generando un número aleatorio *n* y calculando un $M' = n^e M$. Matías presenta el mensaje para firmar a Alicia, quien no viendo ningún problema accede a firmarlo dando como resultado la firma *F'*. A partir de este momento Martín ya tiene la firma de Alicia en el mensaje original *M* ya que $F = \frac{F'}{n} \bmod N = \frac{(M')^d}{n} \bmod N = \frac{(n^e M)^d}{n} \bmod N$, haciendo $F^n = \frac{((n^e M)^d)^e}{n} = \frac{n^{ede} M^{ed}}{n^e} = \frac{n^e M}{n^e} = M$.

Si bien puede parecer un tanto enrevesado, este mismo ataque se puede realizar enviando simplemente el mensaje a un usuario, éste al descifrar el mensaje encontrará una serie ininteligible de símbolos. Probablemente borrará el mensaje que irá a parar en muchos casos a una carpeta de reciclaje o bien según el tipo de Sistema Operativo puede recuperarse mediante una instrucción de reposición del fichero. Si un atacante consigue hacerse con el mensaje borrado podrá recuperar el mensaje original siguiendo los pasos anteriores.

Ataque por módulo común.

La solución más evidente para evitar la generación de un gran número de módulos *N* es la asignación de módulos a grupos de usuarios en lugar de a usuarios particulares y la asignación de pares e_i, d_i a cada usuario en particular. De esta manera la clave secreta de cada usuario sería $\langle N, d_i \rangle$ y la clave pública $\langle N, e_i \rangle$ Sin embargo, esto presenta graves problemas de seguridad, ya que permite que cualquier usuario del grupo obtenga las claves del

resto sin necesidad de factorizar *N*, y, lo que es peor, que un potencial espía obtenga un mensaje común enviado a dos usuarios del grupo.

La obtención de la clave secreta de otro usuario es sencilla. El usuario *A* puede utilizar sus propias claves para factorizar *N*, una vez hecho esto, puede obtener la clave secreta de *B* a partir del conocimiento de su clave pública y de la factorización de *N*.

El proceso de obtención del mensaje *M* enviado a dos usuarios diferentes *i* y *j* se basa en el hecho de que las claves públicas e_i y e_j son valores relativamente primos, con lo cual existirán valores *x* e *y* tales que $xe_i + ye_j = 1$. Aplicando el algoritmo extendido de Euclides se obtienen estos valores y podemos recuperar el mensaje[PAS98].

$$M \bmod N = C_i^x C_j^y \bmod N = M^{xe_i} M^{ye_j} \bmod N = M^{xe_i + ye_j} \bmod N = M \bmod N$$

Ataque cíclico.

Se basa en el hecho de que el sistema RSA es un grupo multiplicativo con un número finito de elementos. El ataque consiste en realizar cifrados sucesivos del cifrado inicial hasta obtener un cifrado igual al inicial, en cuyo caso escogiendo el paso anterior obtenemos el mensaje original.

6.7.3.7 Ataque del cumpleaños (Merkle-Hellman)[PAS98].

Se trata de una adaptación del ataque de Merkle-Hellman de 1981 para romper el DES cifrado con doble clave. Consiste en ir cifrando un mensaje con claves escogidas aleatoriamente hasta que se obtiene una coincidencia.

Sea (*e, N*) la clave pública del RSA, *d* la clave privada y *M* el mensaje a cifrar. Se escogen dos valores aleatorios *i, j* que se utilizarán como clave tales que $1 < i, j < N$ con $i \neq j$. Se calculan los cifrados con ambas claves que denominaremos C_i y C_j. Si se cumple que $C_i = C_j$ tenemos que $M^i \bmod N = M^j \bmod N$ con lo que tenemos que $M^{i-j} = 1 \bmod N$. Aplicando el Teorema Chino del Resto sabemos que la anterior igualdad se cumple siempre que se cumplan las siguientes igualdades ya que $N = p.q.$

$$M^{i-j} = 1 \bmod p$$

$$M^{i-j} = 1 \bmod q$$

Con lo que tenemos por el Teorema de Fermat que $i - j = k_1 p = k_2 p$. Sea $\gamma = mcm(p-1, q-1)$ con lo que podemos afirmar que $i - j = k\gamma \Rightarrow i = j \bmod \gamma$.

El atacante calcula ahora $w = k_3\gamma = \dfrac{i-j}{mcd(e, i-j)}$. Por la definición de *w* tenemos que *w* y *e* son relativamente primos con lo que existen *t* y *s* tales que $sw + te = 1$ y por lo tanto se verifica que $te = 1 \bmod \gamma$, con lo que tenemos que *t* es una de las posibles claves secretas y eso permite al atacante romper el sistema.

Este ataque es fácilmente evitable escogiendo los factores primos *p* y *q* de forma que sean primos fuertes.

Ataque de Wiener a exponentes bajos.

Uno de los principales problemas del RSA es la lentitud del proceso de cifrado, esto es consecuencia de la dificultad de realizar la operación de exponenciación modular. Esta operación requiere un tiempo proporcional al $\log_2 d$. Una de las formas de acelerar este proceso sería la adopción de un valor pequeño de *d*. Sin embargo Wiener demostró que esto era una practica poco recomendable ya que permitía romper el criptosistema[BON99].

El ataque se basa en que si tenemos una clave pública $\langle N, e \rangle$ con $N = p.q$, $q < p < 2q$ y $d < \frac{1}{3} N^{\frac{1}{4}}$, es relativamente fácil recuperar la clave *d*.

La demostración se basa en la utilización de fracciones continuas. Ya que $e.d = 1 \bmod \phi(N)$ tenemos que existe un *k* tal que $ed - k\phi(N) = 1$ con lo que, dividiendo todos los términos por $d\phi(N)$ se cumple que $\left| \frac{e}{\phi(N)} - \frac{k}{d} \right| = \frac{1}{d\phi(N)}$. Al ser *p* y *q* primos tenemos que $\phi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$. Pero como sabemos que $q < p < 2q$ y que al ser $N = p.q$ entonces tenemos que $q < \sqrt{N}$ con lo que $p + q - 1 < p + q < 2q + q = 3q < 3\sqrt{N}$, con lo que tenemos que $|N - \phi(N)| < 3\sqrt{N}$.

Si utilizamos *N* como una aproximación de $\phi(N)$ tenemos que $\left| \frac{e}{\phi(N)} - \frac{k}{d} \right| \approx \left| \frac{e}{N} - \frac{k}{d} \right|$ y sumando y restando $k\phi(N)$ en el numerador obtenemos $\left| \frac{e}{N} - \frac{k}{d} \right| = \left| \frac{ed - k\phi(N) + k\phi(N) - kN}{Nd} \right|$ como $ed - k\phi(N) = 1$ obtenemos despejando en la ecuación anterior $\left| \frac{ed - k\phi(N) + k\phi(N) - kN}{Nd} \right| = \left| \frac{1 - k(N - \phi(N)}{ND} \right| \le \left| \frac{3k\sqrt{N}}{Nd} \right| = \frac{3k}{d\sqrt{N}}$ ya que $|N - \phi(N)| < 3\sqrt{N}$ como habíamos visto anteriormente. Pero inicialmente habíamos supuesto que $d < \frac{1}{3} N^{\frac{1}{4}}$ y ya que tenemos que $ed - k\phi(N) = 1$ obtenemos $k\phi(N) = ed - 1 < ed$ y ya que $e < \phi(N)$ tenemos que $k\phi(N) < d\phi(N) \Rightarrow k < d < \frac{1}{3} N^{\frac{1}{4}}$ con lo que obtenemos $\left| \frac{e}{N} - \frac{k}{d} \le \right| \frac{3k}{d\sqrt{N}} < \frac{3d}{d\sqrt{N}} < \frac{N^{\frac{1}{4}}}{dN^{\frac{1}{2}}} = \frac{1}{dN^{\frac{1}{4}}} < \frac{1}{3d^2} < \frac{1}{2d^2}$. El número de fracciones $\frac{k}{d}$ con $d < N$ aproximándose a $\frac{e}{N}$ está cercano a $\log_2 N$ con lo que lo único que hay que hacer es calcular las soluciones convergentes a $\frac{e}{N}$ y una de ellas será igual a $\frac{k}{d}$, a partir de aquí podemos recuperar *d*.

Ataques basados en el conocimiento parcial de la clave.

Este ataque fue presentado en el AsiaCrypt de 1998 por Boneh, Durfee y Frankel [BON98] y se basa en el conocimiento parcial de la clave privada obtenido mediante algún tipo de ataque, bien sea por hardware o por un ataque por tiempo. Según los autores si el

exponente público *e* es bajo, con una cuarta parte de los bits de la clave, es posible recuperar la clave completa. Si el valor de *e* es mucho mayor, se necesitan la mitad de los bits para reconstruir la clave. La diferencia fundamental entre este ataque y el anterior es que éste no requiere para su éxito limitar el tamaño de *d,* sino simplemente el conocer parte de la clave independientemente de su tamaño. Suponemos que *N = p.q* es un modulo RSA de *n* bits con $\sqrt{N}/2 < q < p < 2\sqrt{N}$, $1 \le e, d \le \phi(N)$ y $e.d \equiv 1 \bmod \phi(N)$. y que siempre que hablemos de los *b* bits más significativos de *d* estamos hablando de los *b* bits de la izquierda. En el caso de que *e* sea pequeño existe un algoritmo tal que dados los $n/4$ bits menos significativos de *d* es capaz de calcular *d* en un tiempo polinomial en función de *n* y *e*. Los autores afirman que el tiempo de ejecución en este caso es proporcional a $e \log_2 e$. Para valores de *e* cercanos a $\sqrt{N}$ es necesario conocer la mitad de los bits más significativos. En la referencia [BON98] se presenta un estudio del ataque para diversos tamaños de *e* y diferentes posiciones de la porción de clave conocida.

Ataque por introducción de fallos de hardware.

El ataque por introducción de fallos es una seria amenaza para las implementaciones del RSA en hardware. La parte computacionalmente menos eficiente en una firma RSA es la exponenciación modular. Sea *M* un mensaje a cifrar, por cuestiones de eficiencia, se suele implementar la exponenciación calculando exponenciaciones parciales módulo *q* y *p*. Es decir, sea $C = M^e \bmod N$ y *a* y *b* dos números que cumplen $\begin{cases} a \equiv 1 \bmod p \\ a \equiv 0 \bmod q \end{cases}$ y $\begin{cases} b \equiv 0 \bmod q \\ b \equiv 1 \bmod q \end{cases}$, se calculan $C_1 = M^e \bmod p$ y $C_2 = M^e \bmod q$, con lo que, mediante la utilización de los números *a* y *b* calculados anteriormente y la utilización del teorema chino del resto, podemos obtener $C = aC_1 + bC_2 \bmod N$.

Supongamos que una vez generado el fallo de hardware, éste solo afecta a C_1 o a C_2, pero no a ambos, en nuestro caso supongamos que el error solo afectó a C_1. El calculo final de *C* que llamaremos C^* se calcula como $C^* = aC_1^* + bC_2^*$. Ya que $C_2 = C_2^*$ podemos hacer $C - C^* = (aC_1 + bC_2) - (aC_1^* + bC_2^*) = a(C_1 - C_1^*)$, si hacemos la suposición, bastante probable, de que $C_1 - C_1^*$ no es divisible por *p* tenemos que el $\text{mcd}(C - C^*, N) = \text{mcd}(a(C_1 - C_1^*), N) = q$, con lo que obtenemos la factorización de *N*.

Ataque por tiempo.

Sea $M = C^d \bmod N$, si tenemos en cuenta que el proceso de exponenciación modular del RSA se realiza de la forma que se explica en el apartado anterior, el ataque se basaría en escoger valores de *c* cercanos a *p* o *q* y determinar, mediante mediciones de tiempo, si el valor supuesto es mayor o menor que el correcto.

Criptoanálisis del método de Hill.

El método de Hill, aunque muy difícil de romper con ataques basándose únicamente en texto cifrado, es fácilmente forzable con un ataque con texto conocido. Supongamos que el oponente ha podido descubrir el valor de *m* utilizado, siendo *m* el grado de la matriz de

permutación. Supongamos que se dispone de al menos *m* pares de tuplas $x_j = (x_{1,j}, x_{2,j}, .., x_{m,j})$ e $y_j = (y_{1,j}, y_{2,j}, ..., y_{m,j})$ con $1 \le j \le m$ y donde $y_j = e_k(x_j)$. Podemos definir dos matrices *X* e *Y* con lo que podemos plantear el sistema *Y=X.K*, si se cumple que la matriz Y es invertible podemos calcular $K = X^{-1}.Y$ y por lo tanto romper el sistema, en caso contrario, necesitaremos obtener otro conjunto de *m* tuplas.

La Inteligencia Artificial y el criptoanálisis.

En los ochenta, y coincidiendo con el auge de las Inteligencia Artificial, los sistemas expertos y los lenguajes deductivos como el prolog se intentan aplicar técnicas de IA al criptoanálisis. Uno de los primeros trabajos al respecto es el de Carroll y Martin [CAR86]. En este trabajo los autores utilizan un sistema experto para realizar el criptoanálisis de sistemas de sustitución. Para ello, los autores asignan un peso a cada símbolo evaluado como texto fuente y generan una base de datos con combinaciones de letras ilegales, terminaciones y palabras de una letra imposibles. En función de los pesos se hace una estimación inicial del texto fuente, el siguiente paso es eliminar las combinaciones imposibles y presentar al usuario una posible solución a un carácter determinado, éste acepta la solución o la rechaza y en el caso de aceptarla se regeneran las bases de datos para contemplar la nueva asunción. Este proceso se realiza continuamente hasta encontrar la solución.

El criptoanálisis de sistemas de sustitución polialfabética es realizado por los autores determinando el número de alfabetos con dos medidas, la covarianza y el índice de coincidencia. Esta medidas son complementarias ya que el índice de coincidencia es más útil para determinar la existencia de pocos alfabetos, hasta cinco en general a partir de los cinco la determinación es bastante ambigua, en cambio el análisis de la covarianza es malo para pocos alfabetos y mejor que el índice de coincidencia para un número superior a los cinco.

Una vez determinado el número de alfabetos se divide el texto en los alfabetos correspondientes y se les aplica el proceso inicial. Los autores aseguran un índice de resultados muy bueno a costa de un rendimiento muy malo. La causa del mal rendimiento es debido principalmente a la utilización del prolog como lenguaje de resolución del problema, lenguaje que nunca se ha caracterizado por su rendimiento.

Posteriormente Carroll y Robbins mejoran el sistema anterior [CAR87]. En este caso utilizan C como lenguaje de implementación. En este caso se utilizan el proceso se basa en asignar pesos a las letras y eliminar combinaciones de digramas y trigramas inexistentes en el idioma inglés. En este caso el proceso es completamente automático y permite encontrar correctamente un 60% de los caracteres de un texto cifrado con el método de Vigenere examinando 1357 símbolos del texto cifrado.

Un avance más significativo fue el realizado por King [KIN92][KIN94], el cual utilizando técnicas de relajación probabilística consigue unos resultados muy buenos con algoritmos de sustitución mono y polialfabeticos respectivamente.

Una revisión y actualización de las técnicas utilizadas en [CAR87] se presenta en [JAK95] el cual presenta una versión iterativa en la que partiendo de un valor probable de la clave, ésta se va actualizando continuamente a partir de una matriz de digramas. Esta matriz va tomando los valores en función de la clave y se actualiza a medida que ésta cambia. Posteriormente se compara el valor de la suma de las diferencias de la matriz con una matriz en la que se almacenan las probabilidades de la aparición de digramas en el lenguaje correspondiente.

Los algoritmos genéticos y el criptoanálisis.

La utilización de los algoritmos genéticos como herramienta criptoanalítica aparece documentada por primera vez en un artículo de la revista Cryptologia de Enero de 1993[SPI93]. Los autores presentan en este trabajo el criptoanálisis de algoritmos de sustitución simple. Para ello utilizan un esquema clásico de algoritmo genético con cruce en un punto, pero utilizando el alfabeto como esquema de representación de los genes. El proceso que sigue el algoritmo para la obtención de la función de aptitud es el siguiente:

1) Se genera una clave para descifrar el texto.
2) Se realiza un análisis de frecuencia del texto resultante a nivel de carácter y de digramas.
3) Las frecuencias obtenidas se comparan con las frecuencias estándares del inglés.
4) Se calcula por último la función de aptitud como:

$$F_{aptitud} = \left(1 - \sum_{i=1}^{26}\left\{\left|f_e(i) - f_m(i)\right| + \sum_{i=1}^{26}\left|f_{ed}(i) - f_{md}(i)\right|\right\}/4\right)^8$$

donde f_e es la frecuencia estándar del lenguaje, f_m es la frecuencia medida, f_{ed} la frecuencia estándar de los digramas y f_{md} la frecuencia obtenida de los digramas.

El algoritmo genético empleado sigue los esquemas clásicos con mutación aplicada a los hijos y promedios de mutación bajos, entre 0,05 y 0,4, obteniendo los mejores resultados con los valores inferiores de mutación. Los resultados fueron más que alentadores, si bien no encontraba la clave correcta en todos los casos, la convergencia no está nunca garantizada en un algoritmo genético, el resultado, si se encontraba, lo hacía después de examinar un conjunto muy reducido de claves.

Otras heurísticas.

La utilización de otros métodos de optimización heurística ha sido continua, pero en general todas las implementaciones realizadas se han concentrado en métodos sencillos. Un caso aparte es el estudio realizado en [CAS00]. En este estudio se intenta utilizar las redes neuronales para criptoanalizar el MD5, sin embargo, tal como reconocen los autores, los resultados fueron prácticamente nulos.

En [FOR93] se aplica el recocido simulado (simulated annealing) a la resolución de cifrados monoalfabéticos con unos resultados más bien moderados.

En general podemos concluir que la utilización de técnicas de inteligencia artificial y técnicas heurísticas ha sido muy poco explotada. El trabajo más completo al que hemos podido acceder es sin duda el de Andrew John Clark [CLA98] que muy amablemente nos cedió una copia del mismo. Sin embargo, se centra en el criptoanálisis de métodos sencillos al igual que la mayoría de los trabajos comentados hasta ahora. La dificultad inherente a la utilización de los métodos derivados de la inteligencia artificial es sin duda el hecho de que no

son adecuados para la obtención de debilidades del sistema, sino que simplemente permiten la explotación de estas debilidades. No se conocen métodos que permitan obtener debilidades de los sistemas que no sean los métodos de ataque típicos deterministas que en la mayoría de los casos son más eficientes que sus implementaciones mediante técnicas de inteligencia artificial. La única razón para utilizar heurísticas es cuando la debilidad obtenida es lo suficientemente difusa como para su implementación mediante un algoritmo determinista. Tal es el caso de la mayoría de los problemas utilizados como base en los sistemas de clave pública. Es quizás en estos en los que las nuevas heurísticas tienen una mayor probabilidad de éxito.

Tendencias.

Las tendencias actuales en criptoanálisis se dividen en dos grandes bloques, potenciar de alguna forma los ataques existentes o evaluar nuevas formas de ataque mediante métodos no probados por ahora, en el primer caso está la búsqueda de esquemas de factorización y la utilización de ataques estadísticos, en el segundo caso la utilización de técnicas de Inteligencia Artificial.

Búsqueda de esquemas de factorización.

Entendemos como factorizar un número el proceso de hallar un conjunto de números primos cuyo producto dé como resultado el número inicial. Calcular el resultado de multiplicar dos números primos es sencillo, por contra factorizar el número resultante es difícil. Esta es la base del algoritmo RSA.

El problema de la factorización se considera difícil desde el punto de vista de la teoría de la complejidad, sin embargo no ha sido probado que lo sea, es posible que en un futuro cercano se demuestre que efectivamente es difícil o bien se encuentre un método que permita factorizar grandes números de una forma sencilla. Hoy por hoy los métodos más rápidos de factorización que se conocen públicamente son la criba cuadrática, la criba cuadrática polinomial múltiple y la criba de campos numéricos. Estos métodos se han utilizado para factorizar números de hasta 150 dígitos. En el apéndice pueden verse algunos de los tests de primalidad y métodos de factorización actuales.

Computación cuántica.

La aparición de los ordenadores cuánticos pueden representar el fin de la criptografía de clave pública tal como la conocemos ahora[SCH00]. La mayoría de los algoritmos utilizados en los criptosistemas de clave pública se basan en dos problemas, la factorización de números y el cálculo del logaritmo discreto. La dificultad de ambos problemas puede verse drásticamente reducida con la aparición de los ordenadores cuánticos. En un ordenador tradicional, el proceso de factorizar un número N con $\log_{10} N$ dígitos necesitaría del orden de $e^{(\log_{10} N)^{\frac{1}{3}}}$, con un ordenador cuántico el número de pasos sería del orden de $(\log_{10} N)^3$[COR00].

Métodos estadísticos.

La estadística es la reina actual en el criptoanálisis. Históricamente los métodos estadísticos son los que mejor resultado han dado y seguramente seguirán haciéndolo durante mucho tiempo. Se trata de una herramienta de valor incalculable para obtener información sobre debilidades en un sistema y en ese apartado no existe ninguna otra que haya dado tan buenos resultados.

Algoritmos de optimización heurística.

En estos métodos se pretende explotar las características de determinados algoritmos de optimización heurística para utilizarlos en el criptoanálisis. Principalmente se han utilizado los algoritmos genéticos y el recocido simulado. Estos algoritmos simulan el funcionamiento de la reproducción natural para resolver problemas que son considerados difíciles mediante métodos más clásicos, tales como el problema de la mochila y el del viajante. El principal inconveniente de estos métodos es el de no ser deterministas, es decir, no garantizan la convergencia y por lo tanto la obtención de un resultado.

El éxito o fracaso en estos métodos se debe en gran medida a la obtención de una función de aptitud adecuada. Esta función en el caso del criptoanálisis debe ser una debilidad inherente al sistema que sea difícilmente explotada por otros métodos. Por el momento solo se han realizado pruebas satisfactorias con criptoanálisis de métodos *sencillos*, queda por ver su utilidad práctica frente a algoritmos modernos, mucho más *fuertes* [CLA96][CLA97][MAT93][FOR93][SPI93][SPI93b][RUB94].