55. Criptografía: historia. Teoría de números. Sistemas de clave pública. Sistemas de clave privada.

Antonio Villalón Huerta
Colegiado número 00033

Resumen: *En este capítulo vamos a profundizar relativamente en los aspectos relacionados con la criptografía en los sistemas de información actuales; tras repasar los conceptos básicos relacionados con esta ciencia, haciendo especial hincapié en sus bases matemáticas, comentaremos aspectos y detalles relacionados con diferentes aproximaciones al cifrado: las clásicas (meramente informativas), la cifra de clave privada y la cifra de clave pública. Para acabar, hablaremos brevemente de aspectos adyacentes al cifrado pero íntimamente ligados a este, como las funciones resumen o la esteganografía.*

1 Introducción

En el mundo físico, si una universidad quiere proteger los expedientes de sus alumnos los guardará en un armario ignífugo, bajo llave y vigilado por guardias, de forma que sólo las personas autorizadas puedan acceder a ellos para leerlos o modificarlos; si queremos proteger nuestra correspondencia de curiosos, simplemente usamos un sobre; si no queremos que nos roben dinero, lo guardamos en una caja fuerte... Lamentablemente, en un sistema de información no disponemos de todas estas medidas que nos parecen habituales: la principal (podríamos decir **única**) forma de protección va a venir de la mano de la criptografía. El cifrado de los datos nos va a permitir desde proteger nuestro correo personal para que ningún curioso lo pueda leer, hasta controlar el acceso a nuestros archivos de forma que sólo personas autorizadas puedan examinar (o lo que quizás es más importante, modificar) su contenido, pasando por proteger nuestras claves cuando conectamos a un sistema remoto o nuestros datos bancarios cuando realizamos una compra a través de Internet. En este capítulo vamos a intentar dar unas bases teóricas

mínimas sobre términos, algoritmos, funciones... utilizadas en ese tipo de aplicaciones. Para más referencias es imprescindible la obra (Schneier, 1994); otras publicaciones interesantes son (Menezes, 1996), (Denning, 1983), (Salomaa, 1990) y, para temas de criptoanálisis, (US Army, 1990). Un texto básico para aquellos que no disponen de mucho tiempo o que sólo necesitan una perspectiva general de la criptografía puede ser (Caballero, 1996); el presente capítulo ha sido extraído en gran parte de (Villalón, 2002).

La criptología (del griego *krypto* y *logos*, estudio de lo oculto, lo escondido) es la ciencia (hemos de dejar patente que la Criptología es una ciencia, aunque en muchos lugares aún se la considera un arte: por ejemplo, en el Diccionario de la Real Academia de la Lengua Española) que trata los problemas teóricos relacionados con la seguridad en el intercambio de mensajes en clave entre un emisor y un receptor a través de un canal de comunicaciones (en términos informáticos, ese canal suele ser una red de computadoras). Esta ciencia está dividida en dos grandes ramas: la **criptografía**, ocupada del cifrado de mensajes en clave y del diseño de criptosistemas (hablaremos de estos más adelante), y el **criptoanálisis**, que trata de descifrar los mensajes en clave, rompiendo así el criptosistema. En lo sucesivo nos centraremos más en la criptografía y los criptosistemas que en el criptoanálisis, ya que nos interesa, más que romper sistemas de cifrado (lo cual es bastante complicado cuando trabajamos con criptosistemas serios), el saber cómo funcionan estos y conocer el diseño elemental de algunos sistemas seguros.

La criptografía es una de las ciencias consideradas como más antiguas, ya que sus orígenes se remontan al nacimiento de nuestra civilización. Su uso original era el proteger la confidencialidad de informaciones militares y políticas, pero en la actualidad es una ciencia interesante no sólo en esos círculos cerrados, sino para cualquiera que esté interesado en la confidencialidad de unos determinados datos: actualmente existe multitud de software y hardware destinado a analizar y monitorizar el tráfico de datos en redes de computadoras; si bien estas herramientas constituyen un avance en técnicas de seguridad y protección, su uso indebido es al mismo tiempo un grave problema y una enorme fuente de ataques a la intimidad de los usuarios y a la integridad de los propios sistemas. Aunque el objetivo original de la criptografía era mantener en secreto un mensaje, en la actualidad no se persigue únicamente la privacidad o confidencialidad de los datos, sino que se busca además garantizar la autenticidad de los mismos (el emisor del mensaje es quien dice ser, y no otro), su integridad (el mensaje que leemos es el mismo que nos enviaron) y su no repudio (el emisor no puede negar el haber enviado el mensaje).

Podemos dividir la historia de la criptografía en tres periodos fundamentales; hasta mediados de siglo, tenemos la criptografía precientífica, considerada no una ciencia sino más bien un arte. En 1949, Shannon logró cimentar la criptografía sobre unas bases matemáticas (Shannon, 1949), comenzando el período de la criptografía científica. Poco

más de diez años después se comenzó a estudiar la posibilidad de una comunicación secreta sin que ambas partes conocieran una clave común (hasta ese momento la existencia de dicha clave era la base de toda la seguridad en el intercambio de información), de forma que esos estudios dieron lugar a diversos artículos sobre el tema durante la década de los setenta ((Ellis, 1970), (Cocks, 1973), (Williamson, 1974), (Williamson, 1976)...). Finalmente, en 1976 Diffie y Hellman publicaron sus trabajos sobre criptografía de clave pública (Diffie, 1976), dando lugar al período de criptografía de clave pública, que dura hasta la actualidad.

2 Teoría de números

La criptografía es una ciencia rodeada desde la antigüedad de un halo de misterio: secretos, ocultación, espías... No obstante, y a pesar de las infinitas historias que se podrían contar relacionadas con los sistemas de cifra, es necesario recordar que su base es puramente matemática, y que está estrechamente relacionada con otras ciencias como la estadística, la teoría de la complejidad, o la teoría de números. En este punto vamos a presentar unas bases matemáticas elementales para poder comprender diferentes aspectos de la criptografía que iremos tratando a lo largo de todo el capítulo; para profundizar más en estas bases matemáticas y obtener referencias adicionales podemos consultar (como casi siempre al hablar de criptografía) la obra (Schneier, 1994).

2.1 Números primos

Se denomina **número primo** a cualquier entero mayor que 1 divisible únicamente por él mismo y por la unidad: estos son sus únicos factores. Ejemplos de números primos son 2, 3 o 5, aunque en criptografía (sobre todo en la de clave pública) es únicamente útil la utilización de números primos extremadamente grandes, de 512 bits o incluso más, como 2^{756839}-1; el conjunto de números primos es obviamente infinito.

Se dice que dos números *a* y *b* son **relativamente primos** si no comparten entre sí más factores que la unidad; un número primo es relativamente primo al resto de números, excepto a sus múltiplos. Otra forma de decir que dos números son relativamente primos es que su máximo común divisor sea la unidad *(mcd(a,b)=1)*, definiendo el **máximo común divisor** de dos números como el número más grande que divide a ambos. Una forma habitual de calcular el máximo común divisor de dos números es mediante el algoritmo de Euclides (que no lo inventó realmente, sino que sólo lo publicó en torno al año 300 a.C.)

2.2 Aritmética modular

Definimos la relación de **congruencia** módulo *p*, denotada por $a \equiv b \bmod p$, si se cumple que $a=b+k \cdot p$, para un entero dado *p* (dicho de otra forma, si *a-b* es múltiplo de *p*). En esta relación, *b* se denomina **residuo** de $a \bmod p$, y se dice que *a* es **congruente** con $b \bmod p$. Al conjunto de enteros de *0* a *p-1* se le denomina conjunto completo de residuos módulo *p*: esto significa que para cada entero *a*, su residuo módulo *p* es un número entero entre *0* y *p-1*. La operación $a \bmod p$ se denomina **reducción modular**, y denota el residuo de *a*, de forma que este residuo es un entero entre *0* y *p-1*; por ejemplo, $100=34 \bmod 11$, ya que $100=34+11 \cdot 6$. Como la aritmética entera, la modular cumple las propiedades conmutativa, asociativa y distributiva.

En criptografía es habitual el uso de la aritmética modular debido a que el cálculo de logaritmos discretos y raíces cuadradas $\bmod p$ pueden ser problemas computacionalmente duros; además, la aritmética modular utiliza cálculos que se realizan cómodamente en ordenadores, ya que restringe tanto el rango de valores intermedios calculados como el resultado final: no tenemos que utilizar grandes números (y por tanto, grandes reservas de memoria) para almacenar resultados, ya que su tamaño siempre estará limitado.

Cuando el número *p* al que hemos hecho referencia es primo forma lo que se denomina un Campo de Galois módulo *p*, denotado *GF(p)*, en el que se cumplen las leyes habituales de la aritmética entera, y que es enormemente utilizado en protocolos criptográficos, como veremos a continuación.

2.3 Exponenciación y logaritmo discreto

Muchos sistemas criptográficos utilizan operaciones de potenciación (exponenciación) en Campos de Galois: elevar una base *a* a una potencia *e* módulo *p*:

$$b = a^e \bmod p \qquad (1)$$

Esta potenciación no es más que una serie de multiplicaciones y divisiones, que computacionalmente tienen un coste lineal con *p* (*O(p)*). Existen aceleraciones al algoritmo directo (multiplicar *e-1* veces la base por sí misma y luego efectuar una reducción modular de un número grande), como la realización de multiplicaciones y reducciones modulares más pequeñas. Por ejemplo, si queremos calcular $a^8 \bmod p$, podemos efectuar la operación directa

$$(a \times a \times a \times a \times a \times a \times a \times a) \bmod p$$

que evidentemente tiene un elevado coste, o realizar un cálculo equivalente y computacionalmente más barato:

$$((a^2 \bmod p)^2 \bmod p)^2 \bmod p$$

El problema inverso a la exponenciación es el cálculo del **logaritmo discreto** de un número módulo *p*: encontrar *x* tal que $a \cdot x = b \bmod p$. Mientras que el problema de la exponenciación es relativamente sencillo, el cálculo del logaritmo discreto es generalmente un problema intratable, y de ahí su interés criptográfico.

3 Criptosistemas

Matemáticamente, podemos definir un criptosistema como una cuaterna de elementos *{A, K, E, D}*, formada por:

- Un conjunto finito llamado **alfabeto**, *A*, a partir del cual, y utilizando ciertas normas sintácticas y semánticas, podremos emitir un mensaje en claro (*plain text*) u obtener el texto en claro correspondiente a un mensaje cifrado (*cipher text*). Frecuentemente, este alfabeto es el conjunto de los enteros módulo *q*, Z_q, para un *q* dado.

- Otro conjunto finito denominado **espacio de claves**, *K*, formado por todas las posibles claves, tanto de cifrado como de descifrado, del criptosistema.

- Una familia de aplicaciones del alfabeto en sí mismo, $E: A \rightarrow A$, llamadas **transformaciones de cifrado**. El proceso de cifrado se suele representar como $E(k,a)=c$, donde $k \in K$, $a \in A$ y $c \in A$.

- Otra familia de aplicaciones del alfabeto en sí mismo, $D: A \rightarrow A$, llamadas **transformaciones de descifrado**. Análogamente al proceso de cifrado, el de descifrado se representa como $D(k',c)=m$, donde $k' \in K$, $c \in A$ y $m \in A$.

Muchos autores dividen a su vez un miembro de esta cuaterna, el alfabeto, en dos espacios diferentes: el espacio de mensajes, *M*, formado por los textos en claro que se pueden formar con el alfabeto, y el espacio de cifrados, *C*, formado por todos los posibles criptogramas que el cifrador es capaz de producir. Sin embargo, lo habitual es que tanto el texto en claro como el cifrado pertenezcan al alfabeto, por lo que hemos preferido no hacer distinciones entre uno y otro, agrupándolos en el conjunto para simplificar los

conceptos que presentamos. Así, un criptosistema presenta la estructura mostrada en la figura siguiente:

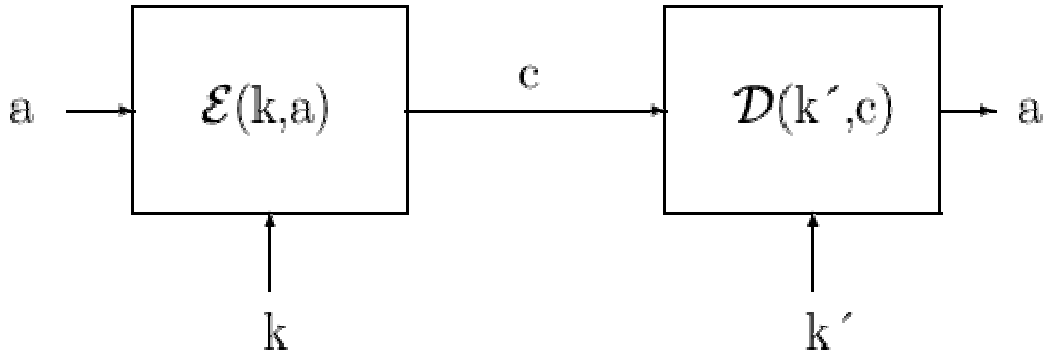

Figura 1. Estructura de un criptosistema.

El emisor emite un texto en claro, que es tratado por un cifrador con la ayuda de una cierta clave, *k*, creando un texto cifrado (criptograma). Este criptograma llega al descifrador a través de un canal de comunicaciones (como hemos dicho antes, para nosotros este canal será habitualmente algún tipo de red), y este convierte el criptograma de nuevo en texto claro, apoyándose ahora en otra clave, (veremos más adelante que esta clave puede o no ser la misma que la utilizada para cifrar). Este texto claro ha de coincidir con el emitido inicialmente para que se cumplan los principios básicos de la criptografía moderna: en este hecho radica toda la importancia de los criptosistemas.

Es obvio, a la vista de lo expuesto anteriormente, que el elemento más importante de todo el criptosistema es el cifrador, que ha de utilizar el algoritmo de cifrado para convertir el texto claro en un criptograma. Usualmente, para hacer esto, el cifrador depende de un parámetro exterior, llamado **clave de cifrado** (o de descifrado, si hablamos del descifrador) que es aplicado a una función matemática irreversible (al menos, computacionalmente): no es posible invertir la función a no ser que se disponga de la clave de descifrado. De esta forma, cualquier conocedor de la clave (y, por supuesto, de la función), será capaz de descifrar el criptograma, y nadie que no conozca dicha clave puede ser capaz de descifrarlo, aún en el caso de que se conozca la función utilizada.

La gran clasificación de los criptosistemas se hace en función de la disponibilidad de la clave de cifrado / descifrado. Existen, por tanto, dos grandes grupos de criptosistemas:

3.1 Criptosistemas de clave secreta

Denominamos **criptosistema de clave secreta** (de clave privada, de clave única o simétrico) a aquel criptosistema en el que la clave de cifrado, *K*, puede ser calculada a partir de la de descifrado, *K'*, y viceversa. En la mayoría de estos sistemas, ambas claves coinciden, y por supuesto han de mantenerse como un secreto entre emisor y receptor: si un atacante descubre la clave utilizada en la comunicación, ha roto el criptosistema.

Hasta la década de los setenta, la invulnerabilidad de todos los sistemas dependía del mantenimiento en secreto de la clave de cifrado. Este hecho presentaba una gran desventaja: había que enviar, aparte del criptograma, la clave de cifrado del emisor al receptor, para que éste fuera capaz de descifrar el mensaje. Por tanto, se incurría en los mismos peligros al enviar la clave, por un sistema que había de ser supuestamente seguro, que al enviar el texto plano. De todos los sistemas de clave secreta, el único que se utiliza en la actualidad es DES (*Data Encryption Standard*, que veremos más adelante); otros algoritmos de clave privada, como el cifrado Caesar o el criptosistema de Vigenère (que serán también brevemente comentados) han sido criptoanalizados con éxito, lo cual da una idea del porqué del desuso en que han caído estos sistemas (con la excepción, insistimos, de DES, que es seguramente el algoritmo de cifra más utilizado en la actualidad). Por si esto no fuera suficiente, el hecho de que exista al menos una clave de cifrado y descifrado entre cada dos usuarios de un sistema haría inviable la existencia de criptosistemas simétricos en las grandes redes de computadores de hoy en día: para un sistema de computación con *N* usuarios, se precisarían $N \cdot (N-1)/2$ claves diferentes, lo cual es obviamente imposible en grandes sistemas. Todos estos motivos han propiciado que el estudio de los cifradores simétricos (excepto DES) quede relegado a un papel histórico.

Los sistemas de cifrado de clave única se dividen a su vez en dos grandes grupos de criptosistemas: por una parte tenemos los cifradores de flujo, que son aquellos que pueden cifrar un solo bit de texto claro al mismo tiempo, y por tanto su cifrado se produce bit a bit, y por otro lado tenemos los cifradores de bloque, que cifran un bloque de bits (habitualmente, cada bloque es de 64 bits) como una única unidad.

3.2 Criptosistemas de clave pública

Como hemos dicho en la introducción, en 1976, Whitfield Diffie y Martin Hellman, de la Universidad de Stanford, demostraron la posibilidad de construir criptosistemas que no precisaran de la transferencia de una clave secreta en su trabajo (Diffie, 1976). Esto motivó multitud de investigaciones y discusiones sobre la criptografía de clave pública y su impacto, hasta el punto que la NSA (*National Security Agency*) estadounidense trató de

controlar el desarrollo de la criptografía, ya que la consideraban una amenaza peligrosa para la seguridad nacional. Esta polémica ha llegado incluso hasta nuestros días, en los que aún siguen surgiendo noticias (no con la misma frecuencia que hace unos años) acerca de las reticencias de muchos gobiernos y organismos de inteligencia con respecto al uso generalizado de la criptografía.

Veamos ahora en que se basan los criptosistemas de clave pública. En éstos, la clave de cifrado se hace de conocimiento general (se le llama **clave pública**). Sin embargo, no ocurre lo mismo con la clave de descifrado (**clave privada**), que se ha de mantener en secreto. Ambas claves no son independientes, pero del conocimiento de la pública no es posible deducir la privada sin ningún otro dato (recordemos que en los sistemas de clave privada sucedía lo contrario). Tenemos pues un par clave pública-clave privada; la existencia de ambas claves diferentes, para cifrar o descifrar, hace que también se conozca a estos criptosistemas como **asimétricos**.

Cuando un receptor desea recibir una información cifrada, ha de hacer llegar a todos los potenciales emisores su clave pública, para que estos cifren los mensajes con dicha clave. De este modo, el único que podrá descifrar el mensaje será el legítimo receptor, mediante su clave privada. Matemáticamente, si E es el algoritmo cifrador y D el descifrador, se ha

$$D(k,E(k',M)) = M \qquad (2)$$

de cumplir que representando M un mensaje, y siendo k y k' las claves de descifrado y cifrado, respectivamente.

4 Criptoanálisis

El criptoanálisis es la ciencia opuesta a la criptografía (quizás no es muy afortunado hablar de ciencias opuestas, sino más bien de ciencias complementarias), ya que si ésta trata principalmente de crear y analizar criptosistemas seguros, la primera intenta romper esos sistemas, demostrando su vulnerabilidad: dicho de otra forma, trata de descifrar los criptogramas. El término `descifrar' siempre va acompañado de discusiones de carácter técnico, aunque asumiremos que descifrar es conseguir el texto en claro a partir de un criptograma, sin entrar en polémicas de reversibilidad y solidez de criptosistemas.

En el análisis para establecer las posibles debilidades de un sistema de cifrado, se han de asumir las denominadas **condiciones del peor caso**:

- El criptoanalista tiene acceso completo al algoritmo de cifrado.
- El criptoanalista tiene una cantidad considerable de texto cifrado.
- El criptoanalista conoce el texto en claro de parte de ese texto cifrado.

También se asume generalmente el **Principio de Kerckhoffs**, que establece que la seguridad del cifrado ha de residir exclusivamente en el secreto de la clave, y no en el mecanismo de cifrado.

Aunque para validar la robustez de un criptosistema normalmente se suponen todas las condiciones del peor caso, existen ataques más específicos, en los que no se cumplen todas estas condiciones. Cuando el método de ataque consiste simplemente en probar todas y cada una de las posibles claves del espacio de claves hasta encontrar la correcta, nos encontramos ante un ataque de fuerza bruta o **ataque exhaustivo**. Si el atacante conoce el algoritmo de cifrado y sólo tiene acceso al criptograma, se plantea un ataque **sólo al criptograma**; un caso más favorable para el criptoanalista se produce cuando el ataque cumple todas las condiciones del peor caso; en este caso, el criptoanálisis se denomina de **texto en claro conocido**. Si además el atacante puede cifrar una cantidad indeterminada de texto en claro al ataque se le denomina de **texto en claro escogido**; este es el caso habitual de los ataques contra el sistema de verificación de usuarios utilizado por Unix, donde un intruso consigue la tabla de contraseñas (generalmente */etc/passwd*) y se limita a realizar cifrados de textos en claro de su elección y a comparar los resultados con las claves cifradas (a este ataque también se le llama **de diccionario**, debido a que el atacante suele utilizar un fichero `diccionario' con los textos en claro que va a utilizar). El caso más favorable para un analista se produce cuando puede obtener el texto en claro correspondiente a criptogramas de su elección; en este caso el ataque se denomina de **texto cifrado escogido**.

Cualquier algoritmo de cifrado, para ser considerado seguro, ha de soportar todos estos ataques y otros no citados; sin embargo, en la criptografía, como en cualquier aspecto de la seguridad, informática o no, no debemos olvidar un factor muy importante: las personas. El sistema más robusto caerá fácilmente si se tortura al emisor o al receptor hasta que desvelen el contenido del mensaje, o si se le ofrece a uno de ellos una gran cantidad de dinero; este tipo de ataques (sobornos, amenazas, extorsión, tortura...) se consideran siempre los más efectivos.

5 Criptografía clásica

En este punto vamos a tratar brevemente algunos criptosistemas históricos obligatorios en cualquier documentación relacionada con la criptología; se trata en cualquier caso de notas que nos pueden ayudar a comprender la importancia del cifrado a lo largo de toda la historia, pero en ningún momento de sistemas robustos utilizados en la actualidad. Cualquier sistema de cifrado clásico, tanto los vistos aquí como otros no comentados (por ejemplo, Enigma), ha sido criptoanalizado con éxito en la actualidad y su rotura es inmediata con la potencia de cálculo existente en nuestros días.

5.1 El cifrado Caesar

El cifrado Caesar (o César) es uno de los más antiguos que se conocen. Debe su nombre al emperador Julio César, que presuntamente lo utilizó para establecer comunicaciones seguras con sus generales durante las Guerras Gálicas.

Matemáticamente, para trabajar con el cifrado Caesar, tomamos el alfabeto Z_m (enteros de módulo *m*). Cuando *a* y *b* son primos entre sí, la aplicación $f(x)=ax+b$, $a\neq0$, recibe el nombre de **codificación módulo m con parámetros a, b**, donde el par *(a,b)* es la clave de este criptosistema. Así, trabajando con el alfabeto inglés (para nosotros resulta conveniente tomar este alfabeto, de uso más extendido en informática que el español; la única diferencia radica en el uso de la letra `*ñ*'), podemos tomar el alfabeto definido por Z_{26}, para lo cual asignamos a cada letra (*a..z*) un entero módulo *26*, de la siguiente forma:

Tabla 1. Alfabeto Caesar.

A=0	B=1	C=2	D=3	E=4	F=5
G=6	H=7	I=8	J=9	K=10	L=11
M=12	N=13	O=14	P=15	Q=16	R=17
S=18	T=19	U=20	V=21	W=22	X=23
Y=24	Z=25				

El cifrado Caesar siempre utiliza la clave *(1,b)*, es decir, siempre tomaremos *a=1*. De esta forma, la anterior aplicación quedará *f(x)=x+b*, lo cual se traduce sencillamente en que para cifrar una letra hemos de tomar su entero correspondiente y sumarle *b* (la clave del criptosistema) para obtener el texto cifrado. Análogamente, y gracias al hecho de que *f(x)* siempre ha de ser biyectiva, independientemente del valor de *b*, para descifrar un texto tomamos la función inversa, definida por $f^{-1}(x)=x-b$. Veamos un ejemplo sencillo, en el que se toma *b=4*: queremos cifrar, con la clave *(1,4)*, la palabra *CESAR*; en primer lugar, tomando el valor de cada letra, tenemos el equivalente numérico de la palabra:

C	E	S	A	R
2	4	18	0	17

A cada uno de los números anteriores le aplicamos la función *f(x)=x+4*, de forma que obtenemos el texto cifrado:

6	8	22	4	21
G	I	W	E	V

Este texto (*GIWEV*) es el resultado de cifrar la palabra *CESAR* con la clave elegida (*b=4*): es lo que enviaríamos al receptor, que conociendo la clave acordada sería capaz de descifrarlo.

Veamos ahora el ejemplo contrario: somos los receptores de un mensaje del que sabemos que ha sido cifrado con la misma clave *(1,4)*, y buscamos descifrar la cadena que nos ha sido enviada, *FVYXYW*. El valor de cada letra es

F	V	Y	X	Y	W
5	21	24	23	24	22

Tomando $f^{-1}(x)=x-4$, tenemos el resultado

1	17	20	19	20	18
B	R	U	T	U	S

Como vemos, retornando cada número al alfabeto inglés obtenemos el texto en claro que nuestro emisor nos ha enviado: *BRUTUS*, equivalente al cifrado *FVYXYW*.

Si en el cifrado de un mensaje obtuviéramos que *f(x)>25* (genéricamente, *f(x)>m-1*), como trabajamos con enteros de módulo *m* deberíamos dividir *f(x)* por *m*, considerando el resto entero como la cifra adecuada. Así, si *f(x)=26*, tomamos *mod(26,26)=0* (el resto de la división entera), por lo que situaríamos una *A* en el texto cifrado.

Es obvio que el cifrado Caesar tiene 26 claves diferentes (utilizando el alfabeto inglés), incluyendo la clave de identidad (*b=0*), caso en el que el texto cifrado y el texto en claro son idénticos. Así pues, no resultaría muy difícil para un criptoanalista realizar un ataque exhaustivo, buscando en el texto cifrado palabras en claro con significado en el lenguaje utilizado. Por este motivo, y por otros muchos, este criptosistema es claramente vulnerable para un atacante, no ofreciendo un nivel de seguridad aceptable en la transmisión de datos confidenciales.

5.2 El criptosistema de Vigenère

El sistema de cifrado de Vigenère (en honor al criptógrafo francés del mismo nombre) es un sistema polialfabético o de sustitución múltiple. Este tipo de criptosistemas aparecieron para sustituir a los monoalfabéticos o de sustitución simple, basados en el Caesar, que presentaban ciertas debilidades frente al ataque de los criptoanalistas relativas a la frecuencia de aparición de elementos del alfabeto. El principal elemento de este sistema es la llamada Tabla de Vigenère, una matriz de caracteres cuadrada, con dimensión 26x26, que se muestra en la tabla siguiente:

Tabla 2. Tableau Vigènere.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

La clave del sistema de cifrado de Vigenère es una palabra de k letras, $k \geq 1$, del alfabeto Z_{26} utilizado anteriormente; esta palabra es un elemento del producto cartesiano $Z_{26} x Z_{26} x Z_{26} x ... x Z_{26}$ (k veces), que es justamente el alfabeto del criptosistema de Vigenère. De esta forma, el mensaje a cifrar en texto claro ha de descomponerse en bloques de k elementos - letras - y aplicar sucesivamente la clave empleada a cada uno de estos bloques, utilizando la tabla proporcionada anteriormente.

Veamos un ejemplo de aplicación del criptosistema de Vigenère: queremos codificar la frase '*La abrumadora soledad del programador*' utilizando la clave '*prueba*'. En primer lugar, nos fijamos en la longitud de la clave: es de seis caracteres, por lo que descomponemos la frase en bloques de longitud seis; aunque el último bloque es de longitud tres, esto no afecta para nada al proceso de cifrado:

laabru madora soleda ddelpr ograma dor

Ahora, aplicamos a cada bloque la clave *prueba* y buscamos los resultados como entradas de la tabla de Vigenère:

laabru	*madora*	*soleda*	*ddelpr*	*ograma*	*dor*
prueba	*prueba*	*prueba*	*prueba*	*prueba*	*pru*
arufsu	*brxhsa*	*igfiea*	*suyoqr*	*exmena*	*sgm*

Por ejemplo, la primera *a* del texto cifrado corresponde a la entrada *(l,p)*, o equivalentemente, *(p,l)*, de la tabla de Vigenère. Finalmente, vemos que el texto cifrado ha quedado *arufsu brxhsa igfiea suyoqr exmena sgm*.

Este método de cifrado polialfabético se consideraba invulnerable hasta que en el siglo XIX se consiguieron descifrar algunos mensajes codificados con este sistema, mediante el estudio de la repetición de bloques de letras: la distancia entre un bloque y su repetición suele ser múltiplo de la palabra tomada como clave.

Una mejora sobre el cifrado de Vigenère fue introducida por el sistema de Vernam, utilizando una clave aleatoria de longitud igual a la del mensaje; la confianza en este nuevo criptosistema hizo que se utilizase en las comunicaciones confidenciales entre la Casa Blanca y el Kremlin, hasta, por lo menos, el año 1987.

6 Criptografía de clave privada

6.1 El criptosistema DES

El DEA (*Data Encryption Algorithm*) o DES (*Data Encryption Standard*) es desde 1977 de uso obligatorio en el cifrado de informaciones gubernamentales no clasificadas (anunciado por el *National Bureau of Standards*, USA). Este criptosistema fue desarrollado por IBM como una variación de un criptosistema anterior, Lucifer, y posteriormente, tras algunas comprobaciones llevadas a cabo por la NSA estadounidense, pasó a transformarse en el que hoy conocemos como DES. Este sistema puede ser implementado tanto en software como en chips con tecnología VLSI (*Very Large Scale Integration*), alcanzando en hardware una velocidad de hasta 50 Mbps. Un ejemplo de implantación hardware puede ser PC-Encryptor, de Eracom, y un ejemplo de implantación en software es DES-LOCK, de la empresa Oceanics.

DES es un sistema de clave privada tanto de cifrado como de descifrado; posee una clave de entrada con una longitud de 64 bits, produciendo una salida también de 64 bits, con

una clave de 56 bits (el octavo bit de cada byte es de paridad), llamada **clave externa**, en la que reside toda la seguridad del criptosistema ya que el algoritmo es de dominio público. Cada trozo de 64 bits de los datos se desordena según un esquema fijo a partir de una permutación inicial conocida como **IP**. A continuación, se divide cada uno de los trozos en dos mitades de 32 bits, que se someten a un algoritmo durante 16 iteraciones. Este algoritmo básico que se repite 16 veces (llamadas **vueltas**), utiliza en cada una de ellas 48 de los 56 bits de la clave (estos 48 bits se denominan clave interna, diferente en cada vuelta); las claves internas se utilizan en un orden para cifrar texto (llamémoslas $K_1,K_2,...,K_{16}$) y en el orden inverso ($K_{16},K_{15},...,K_1$) para descifrarlo. En cada una de las vueltas se realizan permutaciones, sustituciones no lineales (que constituyen en sí el núcleo del algoritmo DES) y operaciones lógicas básicas, como la XOR. La mitad derecha se transfiere a la mitad izquierda sin ningún cambio; también se expande de 32 hasta 48 bits, utilizando para ello una simple duplicación. El resultado final de una iteración es un XOR con la clave interna de la vuelta correspondiente, y esta salida se divide en bloques de 6 bits, cada uno de los cuales se somete a una sustitución en un bloque de 4 bits (**bloque-S**, con un rango *0...63*) dando una salida también de 4 bits (rango decimal *0...15*) que a su vez se recombina con una permutación en un registro con longitud 32 bits. Con el contenido de este registro se efectúa una operación XOR sobre la mitad izquierda de los datos originales, convirtiéndose el nuevo resultado en una salida (parte derecha) de 32 bits; transcurridas las dieciséis vueltas, las dos mitades finales (de 32 bits cada una) se recombinan con una permutación contraria a la realizada al principio (IP), y el resultado es un criptograma de 64 bits.

Aunque no ha sido posible demostrar rigurosamente la debilidad del criptosistema DES, y actualmente es uno de los más utilizados en el mundo entero, parece claro que con las actuales computadoras y su elevada potencia de cálculo una clave de 56 bits (en la que recordemos, reside toda la seguridad del DES) es fácilmente vulnerable frente a un ataque exhaustivo en el que se prueben combinaciones de esos 56 bits. Hay que resaltar que el tamaño inicial de la clave, en el diseño de IBM, era de 128 bits; la razón de la disminución no se ha hecho pública hasta el momento. Por si esto fuera poco, otro factor que ha aumentado las controversias y discusiones acerca de la seguridad de DES son dos propiedades del algoritmo: la **propiedad de complementación**, que reduce el tiempo necesario para un ataque exhaustivo, y la propiedad de las claves débiles, dada cuando el proceso de cifrado es idéntico al de descifrado ($K_1=K_{16}$, $K_2=K_{15},...,K_8=K_9$), que sucede con cuatro claves del criptosistema. Otro secreto de IBM (a instancias de la NSA) es la elección y diseño de las cajas que DES utiliza para el cifrado; no se puede evitar el pensar que el gobierno estadounidense precisaraá un criptosistema con la robustez necesaria para que nadie, excepto ellos, pueda descifrarlo.

A la vista de estos hechos, la idea de que DES no va a seguir siendo el algoritmo de cifrado estándar en las instituciones estadounidenses se va generalizando poco a poco; por tanto, va a ser necesario sustituirlo por otro algoritmo más robusto frente a los ataques. Siguiendo esta línea, Xuejia Lai y James Massey, dos prestigiosos criptógrafos, desarrollaron a finales de la década de los ochenta el algoritmo IDEA (*International Data Encryption Algorithm*), compatible con DES (para aprovechar el gran número de equipos que utilizan este algoritmo), y con una robustez garantizada por la clave de 128 bits que utiliza este cifrador de bloques y las complejas operaciones utilizadas para evitar el éxito de un posible atacante, que van desde técnicas de difusión hasta adiciones módulo 216.

El algoritmo IDEA está siendo ampliamente aceptado en diversas aplicaciones informáticas orientadas a la seguridad de los datos; numerosos programas destinados a trabajar en red utilizan ya este algoritmo como el principal de cifrado.

7 Criptosistemas de clave pública

7.1 El criptosistema RSA

Este sistema de clave pública fue diseñado en 1977 por los profesores del MIT (*Massachusetts Institute of Technology*) Ronald R. Rivest, Adi Shamir y Leonard M. Adleman, de ahí las siglas con las que es conocido. Desde entonces, este algoritmo de cifrado se ha convertido en el prototipo de los de clave pública.

La seguridad de RSA radica en la dificultad de la factorización de números grandes: es fácil saber si un número es primo, pero es extremadamente difícil obtener la factorización en números primos de un entero elevado, debido no a la dificultad de los algoritmos existentes, sino al consumo de recursos físicos (memoria, necesidades hardware... incluso tiempo de ejecución) de tales algoritmos. Se ha demostrado que si n es el número de dígitos binarios de la entrada de cualquier algoritmo de factorización, el coste del algoritmo es $\theta(2n)$, con un tiempo de ejecución perteneciente a la categoría de los llamados problemas intratables.

Veamos el funcionamiento del algoritmo RSA: si un usuario A desea enviar información cifrada, en primer lugar tiene que calcular un par de claves (pública y privada), para lo que ha de elegir aleatoriamente dos números primos grandes (del orden de cien dígitos), p y q, números que se han de mantener en secreto; si llamamos N (N se conoce como **módulo**) al producto $p \cdot q$, el usuario ha de determinar otro entero, d, llamado **exponente privado**, que cumpla

$$\text{mcd } (d, (p-1)x(q-1)) = 1,\ d<N \qquad (3)$$

es decir, *d* y el producto *(p-1)x(q-1)*, que llamaremos **función de Euler** y denotaremos $\varphi(N)$, han de ser primos. Con estos datos, ya tenemos la clave privada del cifrado: el par *(N,d)*; para obtener la clave pública, hallamos el inverso multiplicativo del número *d* respecto de $\varphi(N)$, de la forma $e \cdot d = 1 \bmod \varphi(N)$. Calculado este entero *e*, llamado **exponente público**, la clave pública será el par *(N,e)*.

Una vez el emisor *A* dispone de sus claves pública y privada, podría enviar un mensaje cifrado, que llamaremos *m*, a un posible receptor, mediante la operación $c=m^e \bmod N$ aplicada a cada elemento del mensaje. Cuando el receptor del criptograma desee descifrar el mensaje recibido, ha de realizar la operación $m=c^d \bmod N$ para obtener el texto en claro del mensaje que acaba de recibir.

El sistema RSA ha permanecido invulnerable hasta hoy, a pesar de los numerosos ataques de criptoanalistas; teóricamente es posible despejar d para obtener la clave privada, a partir de la función de descifrado, resultando

$$d= \log_c m \ (\text{mod}(p-1)) \qquad (4)$$

Sin embargo, el cálculo de logaritmos discretos es un problema de una complejidad desbordante, por lo que este tipo de ataque se vuelve impracticable: la resolución de congruencias del tipo $a^x \equiv b(n)$ necesarias para descifrar el mensaje, es algorítmicamente inviable sin ninguna información adicional, debido al elevado tiempo de ejecución del algoritmo. Aunque cuando los factores de *(p-1)* son pequeños existe un algoritmo, desarrollado por Pohlig y Hellman de orden $O(\log^2 p)$, este es otro de los algoritmos catalogados como intratables, ya comentados anteriormente.

7.2 Criptosistema de ElGamal

Durante 1984 y 1985 ElGamal desarrolló un nuevo criptosistema de clave pública basado en la intratabilidad computacional del problema del logaritmo discreto: obtener el valor de *x* a partir de la expresión $y \equiv a^x \pmod p$ es, como hemos visto para el caso de RSA, computacionalmente intratable por norma general.

Aunque generalmente no se utiliza de forma directa, ya que la velocidad de cifrado y autenticación es inferior a la obtenida con RSA, y además las firmas producidas son más largas (¡el doble de largo que el texto original!), el algoritmo de ElGamal es de gran

importancia en el desarrollo del DSS (*Digital Signature Standard*), del NIST (*National Institute of Standards and Technology*) estadounidense.

En el criptosistema de ElGamal, para generar un par clave pública / clave privada, se escoge un número primo grande, *p*, y dos enteros *x* y *a*, $1 \le x \le p-1$, $1 \le a \le p-1$, y se calcula $y=a^x \pmod p$. La clave pública será el número *y*, y la privada el número *x*.

Para firmar un determinado mensaje, el emisor elige un entero aleatorio *k*, $0<k<p-1$, no usado con anterioridad y con la restricción que sea relativamente primo a $(p-1)$, y computa

$$r = a^k \bmod p \tag{5}$$

$$s = [\, k^{-1}(m-xr)] \bmod (p-1) \tag{6}$$

donde k^{-1} es el inverso de $k \bmod (p-1)$, de forma que $k \cdot k^{-1} = 1 \bmod (p-1)$. La firma del mensaje estará entonces formada por *r* y *s*; un potencial receptor puede usar la clave pública y para calcular $y^r r^s \bmod p$ y comprobar si coincide con $a^m \bmod p$:

$$y^r r^s \bmod p = a^m \bmod p \tag{7}$$

El criptosistema de ElGamal tiene una característica determinante que lo distingue del resto de sistemas de clave pública: en el cifrado se utiliza aparte de la clave pública del receptor, la clave privada del emisor.

8 Funciones resumen

Matemáticamente podemos definir las funciones resumen (*hash functions*) como proyecciones de un conjunto, generalmente con un número elevado de elementos (incluso infinitos), sobre un conjunto de tamaño fijo y mucho más pequeño que el anterior; por ejemplo, podemos definir la siguiente función resumen, que va de un conjunto con un número infinito de elementos a otro con únicamente 10:

$$H(x) = x \bmod 10,\ x \in \Re,\ H(x) \in [0,9]$$

Sin embargo, aunque la anterior sea una función resumen en sentido estricto, no es especialmente interesante en aplicaciones criptográficas; para serlo, habría de cumplir los siguientes requisitos:

- La entrada puede ser de un tamaño indeterminado.
- La salida es de un tamaño fijo, varios órdenes de magnitud más pequeño que el anterior.
- Para un cierto *x*, calcular *H(x)* es computacionalmente barato.
- *H(x)* es de un solo sentido.
- *H(x)* no presenta colisiones.

El que una función *hash* sea de un **solo sentido** (lo que se denomina *One-Way hash function*) no implica más que a partir del valor de *H(x)* no puedo obtener el de *x*: no existe, o su cálculo es computacionalmente difícil. Las colisiones en una función resumen se producen cuando para dos entradas diferentes *x* e *y*, *H(x)=H(y)*, y se habla de funciones *hash* **débilmente libres de colisiones** (*weakly collision free*) cuando es computacionalmente imposible encontrar dos elementos *x* e *y* tales que cumplan *H(x)=H(y)*; si aparte de computacionalmente imposible también lo es matemáticamente, se habla de funciones resumen **fuertemente libres de colisiones** (*strongly collision free*).

Una de las aplicaciones criptográficas más importante de las funciones resumen es sin duda la verificación de integridad de archivos; la idea es sencilla: en un sistema del que tengamos constancia que está `limpio' (esto es, que no ha sido troyanizado o modificado de cualquier forma por un pirata) podemos generar resúmenes de todos los ficheros que consideremos clave para el correcto funcionamiento de la máquina y guardar dichos resúmenes - como ya indica su nombre, mucho más cortos que los archivos originales - en un dispositivo de sólo lectura como un CD-ROM. Periódicamente, o cuando sospechemos que la integridad de nuestro entorno ha sido violada, podemos volver a generar los resúmenes y comparar su resultado con el almacenado previamente: si no coinciden, podemos estar seguros (o casi seguros) de que el fichero ha sido modificado.

Para este tipo de aplicaciones se suele utilizar la función resumen **MD5**, diseñada por Ronald Rivest y que viene implementada `de serie' en sistemas operativos Unix, como Solaris o Linux (órdenes como *md5* o *md5sum*):

```
$ echo "Esto es una prueba" >/tmp/salida
$ md5sum /tmp/salida
3f8a62a7db3b276342d4c65dba2a5adf  /tmp/salida
$ echo "Ahora modifico el fichero" >>/tmp/salida
$ md5sum /tmp/salida
1f523e767e470d8f23d4378d74817825  /tmp/salida
$
```

Otra aplicación importante de las funciones resumen es la firma digital de mensajes - documentos - y su marca de tiempo (*timestamping)*; en el primer caso, como los algoritmos de firma digital suelen ser lentos, o al menos más lentos que las funciones *hash*, es habitual calcular la firma digital de un resumen del fichero original, en lugar de hacer el cálculo sobre el propio fichero (evidentemente, de tamaño mayor que su resumen). Con respecto al *timestamping*, las funciones *hash* son útiles porque permiten publicar un resumen de un documento sin publicar su contenido, lo cual permite a una parte obtener un *timestamp* de un documento sin que la autoridad de *timestamp* conozca el contenido del mismo, pero asegurándose la validez del procedimiento en caso de repudio; en ambos casos, tanto en la firma digital como en el *timestamping*, trabajar con el resumen es completamente equivalente a trabajar con el archivo original.

9 Esteganografía

La esteganografía (también llamada **cifra encubierta**, (CESID, 1991)) es la ciencia que estudia los procedimientos encaminados a ocultar la existencia de un mensaje en lugar de ocultar su contenido; mientras que la criptografía pretende que un atacante que consigue un mensaje no sea capaz de averiguar su contenido, el objetivo de la esteganografía es ocultar ese mensaje dentro de otro sin información importante, de forma que el atacante ni siquiera se entere de la existencia de dicha información oculta. No se trata de sustituir al cifrado convencional sino de complementarlo: ocultar un mensaje reduce las posibilidades de que sea descubierto; no obstante, si lo es, el que ese mensaje haya sido cifrado introduce un nivel adicional de seguridad.

A lo largo de la historia han existido multitud de métodos para ocultar información. Quizás los más conocidos hayan sido la tinta invisible, muy utilizada durante la Segunda Guerra Mundial, o las marcas de cualquier tipo sobre ciertos caracteres (desde pequeños pinchazos de alfiler hasta trazos a lápiz que marcan un mensaje oculto en un texto), pero otros mecanismos más extravagantes también han sido utilizados: por ejemplo, afeitar la cabeza de un mensajero y tatuar en el cuero cabelludo el mensaje, dejando después que

el crecimiento del pelo lo oculte; podemos repasar algunos modelos esteganográficos cuanto menos curiosos en (Kahn, 1967).

Con el auge de la informática, el mecanismo esteganográfico más extendido está basado en las imágenes digitales y su excelente capacidad para ocultar información; aunque existen varias formas de conseguirlo (Schyndel, 1994), la más básica consiste simplemente en sustituir el bit menos significativo de cada byte por los bits del mensaje que queremos ocultar; dado que casi todos los estándares gráficos tienen una graduación de colores mayor de lo que el ojo humano puede apreciar, la imagen no cambiará su apariencia de forma notable. Otros elementos donde ocultar información son las señales de audio y video y el propio texto (Bender, 1996); aunque históricamente nunca han estado tan extendidas como la anterior, en los últimos tiempos el interés por los mecanismos de ocultación de información en formatos de audio (principalmente MP3) y video ha ido en aumento. Y no es de extrañar: a nadie se le escapa que con la cantidad de protocolos *peer to peer* de intercambio de archivos (*e-Donkey*, *Morpheus*...) que existen en la actualidad, y que son usados por millones de usuarios para intercambiar ficheros MP3 y DIVX a través de la red, el volumen de información que puede viajar camuflada en los mismos es impresionante. Esto, que a la mayor parte de los mortales nos da un poco igual, es un área de gran interés para las agencias de inteligencia de todo el mundo (muy en especial desde los desgraciados sucesos del 11-S), debido al peligro que entraña el intercambio de información discreto, rápido y efectivo que puede establecerse entre miembros de redes terroristas desde cualquier punto del planeta, sin más que un PC conectado a Internet y un programa cliente de cualquiera de estos protocolos.

Referencias bibliográficas

Bender, W., Gruhl, D., Morimoto, N. y Lu, A. (1996). *Techniques for data hiding*. IBM Systems Journal. 35, 3, 4.

Caballero, P. (1996). *Introducción a la Criptografía*. Ra-Ma.

CESID (1991). *Glosario de términos de Criptología*. Centro Superior de Información de la Defensa.

Cocks, C.C. (1973). *A note on non-secret encryption*. CESG Technical Report.

Denning, D. (1983). *Cryptography and Data Security*. Addison-Wesley.

Diffie, W. y Hellman, M.E. (1976). *New directions in cryptography*. IEEE Transactions on Information Theory. IT-22, 644—654.

Ellis, J.H. (1970). *The possibility of non-secret digital encryption*. CESG Technical Report.

Kahn, D. (1967). *The Codebreakers*. McMillan.

Menezes, A., van Oorschot, P. y Vanstone, S. (1996). *Handbook of Applied Cryptography*. CRC Press.

Salomaa, A. (1990). *Public Key Cryptography*. Springer-Verlag.

Schneier, B. (1994). *Applied Cryptography: Protocols, Algorithms and Source Code in C*. John Wiley & Sons.

Schyndel, R.G., Tirkel, A.Z. y Osborne, C.F. (1994). *A digital watermark*. International Conference on Image Processing. 2, 86-90. IEEE Press.

Shannon, C.E. (1949). *Communication theory of secrecy systems*. Bell Systems Technology Journal, 28, 657—715.

US Army Headquarters (1990). *Basic cryptanalysis*. Technical Report FM-34-40-2.

Villalón, A. (2002). *Seguridad en Unix y Redes*. GNU Free Documentation.

Williamson, M.J. (1974). *Non-Secret encryption using a finite field*. CESG Technical Report.

Williamson, M.J. (1976). *Thoughts on cheaper Non-Secret encryption*. CESG Technical Report.