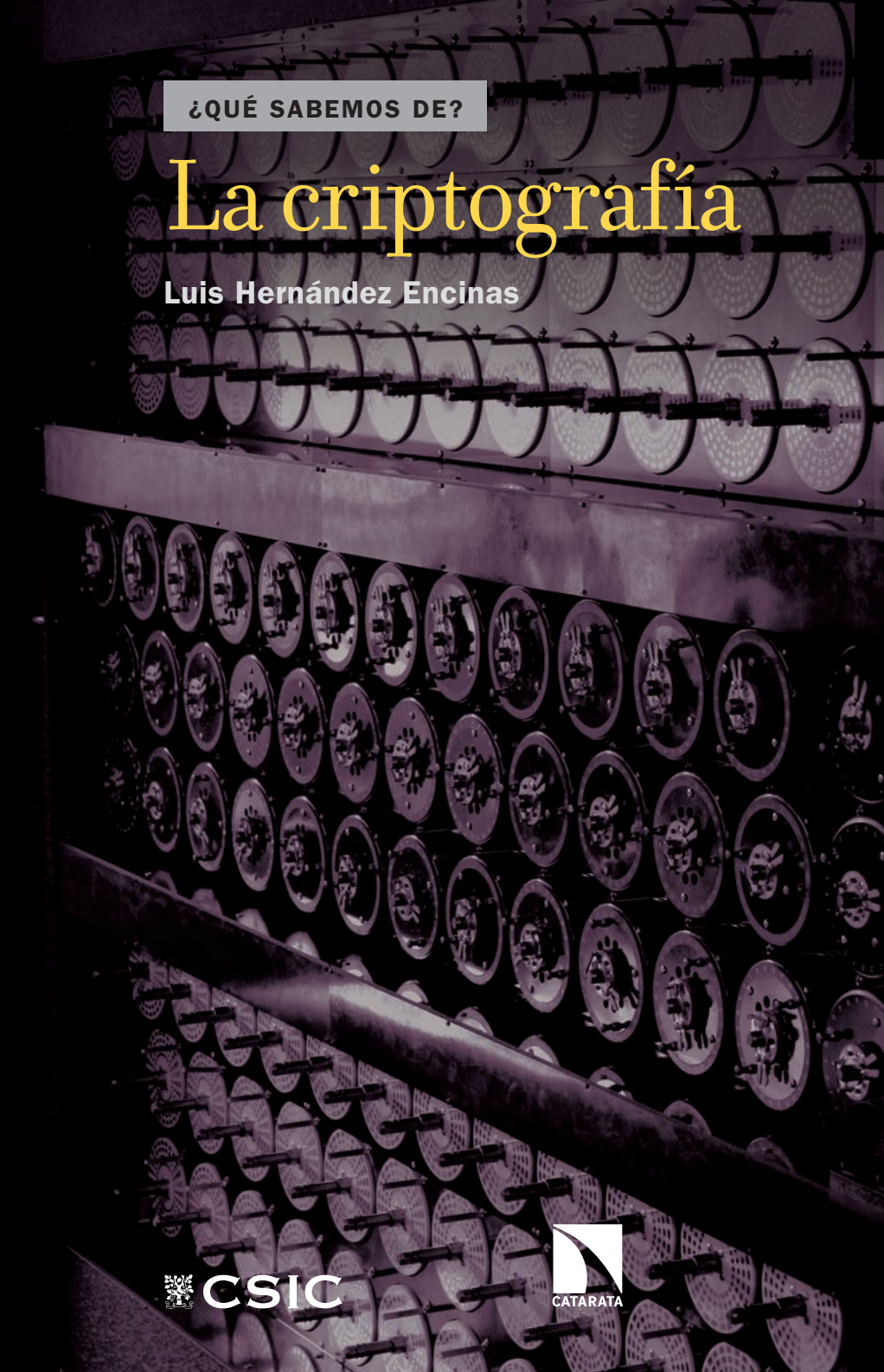


¿QUÉ SABEMOS DE?

La criptografía

Luis Hernández Encinas

CSIC

CATARATA

La criptografía

Luis Hernández Encinas

CSIC

CATARATA

Catálogo general de publicaciones oficiales
http://publicacionesoficiales.boe.es

Gobierno de España · Ministerio de Economía y Competitividad · CSIC · Catarata

Diseño gráfico de cubierta: Carlos Del Giudice
Fotografía de cubierta: © Thinkstock

Fuencarral, 70
28004 Madrid
Tel. 91 532 20 77
Fax. 91 532 43 34
www.catarata.org

ISBN (CSIC): 978-84-00-10045-2
ISBN electrónico (CSIC): 978-84-00-10046-9
ISBN (Catarata): 978-84-9097-107-9
NIPO: 723-16-001-3
NIPO electrónico: 723-16-002-9
Depósito legal: M-2.860-2016
IBIC: PDZ-GPJ

A mi hijo Luis, para quien el RSA
y la ECC han dejado de tener secretos

Índice

Agradecimientos

Quiero agradecer a mi compañeros criptógrafos, Raúl Durán y Víctor Gayoso, sus comentarios de todo tipo para mejorar la redacción y el contenido de esta obra; a Jesús Negrillo por su afán en lograr la perfección a la hora de fotografiar cualquier cosa que se ponga por delante, en especial si son máquinas de cifrado; y a Agustín Martín su tiempo y dedicación, con quien los intercambios de parecer han sido largos e intensos, y cuyo afán en que las cosas sean claras y comprensibles no tiene fin.

También quiero expresar mi gratitud al Centro Criptológico Nacional (CCN) por las facilidades que me ha dado en todo momento para conseguir algunas de las mejores fotos que se incluyen en este libro. Quiero destacar, en especial, la oportunidad que me ha dado de hacer cuantas fotos he querido de su máquina Enigma, que, de no ser así, no podrían mostrarse en esta obra con tanto detalle.

Introducción

La necesidad de guardar secretos

Desde siempre, el hombre ha sentido la necesidad de tener secretos y guardarlos a buen recaudo. Tan solo en algunas situaciones deseaba compartirlos con determinados amigos o aliados, asegurándose de que aquellos no eran conocidos por terceras partes.

Esta necesidad ha hecho que, a lo largo de la historia, el hombre haya aguzado su ingenio con el fin de proteger determinada información, inventando métodos que le permitieran ocultarla o convertirla en secreta ante los que consideraba enemigos. A la vez, debería poder acceder a ella en caso de necesidad o permitir que, bajo determinadas condiciones, sus aliados pudieran conocerla.

Tradicionalmente, existen dos formas de lograr este objetivo. Una de ellas consiste en ocultar el propio hecho de la existencia de un mensaje secreto, de modo que si se desconoce que tal mensaje existe, un adversario no tendrá la preocupación de buscarlo. La otra es la de llevar a cabo diferentes modificaciones o transformaciones en el mensaje para que, si este cae en manos no deseadas, sea imposible conocer la información a que hace referencia. Además, tales transformaciones deben permitir que el mensaje pueda ser

recuperado más adelante, bien por el propietario, bien por el destinatario.

Ocultamiento de mensajes: esteganografía

El método de guardar secretos que consiste sencillamente en ocultar el mero hecho de la existencia de un mensaje se conoce como *esteganografía* (Van Tilborg *et al.*, 2005). El término procede de las palabras griegas *steganos*, que significa "que cubre", "que protege", "cubierto", y *graphein*, que es "escribir". Así pues, la esteganografía trata de cómo escribir un mensaje de modo que quede encubierto u oculto.

Uno de los primeros métodos esteganográficos documentados fue mencionado por Heródoto (484-425 a.C.) en *Las Historias* (Heródoto, 1985). Este señala que Demarato (515-491 a.C.), para avisar a sus conciudadanos griegos de los planes de invasión de Jerjes (519-465 a.C.), allá por el año 480 a.C., limpió la cera de una tablilla, escribió sobre la madera el mensaje y volvió a colocar la cera, de modo que la tablilla parecía estar sin escribir. Al llegar la tablilla a su destino, bastó con quitar la cera para leer el mensaje.

El mismo Heródoto narra otro sistema para la ocultación de un mensaje, posiblemente más curioso que el anterior. En este caso se trataba de cómo Histaiaeo (o Histieo, c.a. 494 a.C.) pidió a Aristágoras de Mileto (finales del siglo VI-principios del siglo V a.C.) que se rebelara contra el rey de Persia. Histaiaeo afeitó la cabeza del que iba a ser el mensajero, escribió el mensaje en su cabeza y cuando le creció el pelo, lo envió a su destino. Al llegar, le volvieron a afeitar la cabeza con lo que el mensaje quedó al descubierto sin que nadie tuviera conocimiento siquiera de la existencia del mismo. Detalles como estos ponen de manifiesto que la medida del tiempo en aquella época no era la misma que la empleada hoy en día.

A lo largo de la historia se han utilizado otros muchos métodos esteganográficos. Así, se sabe que en la antigua China se escribían mensajes sobre seda fina; luego se hacía

una pequeña pelota con la seda, que era envuelta en cera y que el mensajero se tragaba.

Otro método utilizado a lo largo de la historia ha sido el uso de las denominadas "tintas invisibles". En este caso se trata de escribir el mensaje con una tinta que desaparece y que lo vuelve invisible.

Plinio el Viejo (o Gayo Plinio Segundo, 23-79) señaló que la leche de la planta *Tith ymalus* podía utilizarse como tinta invisible, puesto que al secarse se volvía transparente, mientras que si se calentaba reaparecía con un tono marrón (Plinio, 2007). Por su parte, Giovanni Battista della Porta (o Giambattista della Porta, 1535-1615) describió cómo era posible ocultar un mensaje dentro de un huevo cocido. Para ello bastaba con elaborar una tinta a base de alumbre y vinagre y luego escribir en la cáscara del huevo. La tinta atraviesa la cáscara y deja el mensaje en el huevo, de modo que solo es posible leerlo si este se pela.

Otro método para ocultar un mensaje consiste en hacerlo formar parte de otro mensaje, de modo que el primero pase inadvertido dentro del segundo.

Uno de los ejemplos más conocidos de la literatura española es el de los acrósticos de *La Celestina* (De Rojas, 2013), es decir, los versos cuyas letras iniciales forman un mensaje; en este caso, el nombre del autor de la obra: Fernando de Rojas (1470-1541):

> […]
> **F**uertes más que ella por cebo la llevan:
> **E**n las nuevas alas estaba su daño.
> **R**azón es que aplique a mi pluma este engaño,
> **N**o disimulando con los que arguyen;
> **A**sí que a mí mismo mis alas destruyen,
> **N**ublosas e flacas, nacidas de hogaño.
> **D**onde esta gozar pensaba volando,
> **O** yo aquí escribiendo cobrar más honor,
> **D**e lo uno y lo otro nació disfavor:
> **E**lla es comida y a mí están cortando

Reproches, revistas e tachas. Callando
Obstara los daños de envidia e murmuros;
Y así navegando, los puertos seguros
Atrás quedan todos ya, cuanto más ando.
Si bien discernís mi limpio motivo,
[…]

Este método, con bastante mayor sofisticación, aparece con frecuencia en las películas de espías, de modo que para leer un mensaje hace falta un libro o texto de referencia y una guía o clave de lectura. Esta guía suele ser una colección de grupos de tres números que hacen referencia a la página, al número de línea en esa página y a la posición de la palabra en dicha línea. En algunas ocasiones esta forma de ocultar un mensaje se conoce como "canal subliminal".

Más recientemente, la esteganografía ha sido utilizada por los agentes de los servicios secretos de algunos países para enviar información de forma secreta. En esta ocasión, el avance de la tecnología permitía microfilmar documentos, de modo que el microfilme era pegado en cartas ordinarias sustituyendo el punto de una *i*, de una *j* o un punto de fin de frase. Parece ser que el FBI descubrió por primera vez un micropunto en 1941, a partir de un soplo que informaba que debían buscar un pequeño brillo en una carta procedente de agentes alemanes en Latinoamérica.

Hoy en día, el uso de la esteganografía se ha generalizado con el fin de proteger determinados mensajes o información relacionada con estos, aunque se conozca su existencia. Se trata de incluir información adicional al mensaje e íntimamente ligada al mismo de modo que se detecte cualquier modificación del mensaje, en general, para proteger su contenido.

Así, los billetes de banco, cheques o archivos multimedia, por ejemplo, incluyen información no visible que permite protegerlos contra su falsificación o duplicación indebida.

Uno de los usos más sorprendentes para la protección de los billetes de cualquier denominación de euro (y actualmente

de otros muchos billetes de curso legal) es la impresión en cada uno de ellos (a lo largo de todo el billete y en cualquier posición) de una configuración especial de cinco pequeños círculos denominada *eurión* (una forma muy esquemática de la constelación de Orión) con un color determinado en cada billete (figura 1).

FIGURA 1

Representación del eurión en billetes de curso legal.

FUENTE: LUIS HERNÁNDEZ ENCINAS.

Esta configuración es detectada por todas las fotocopiadoras de color modernas[1], de modo que si se intenta fotocopiar un billete que lleve impresa esta configuración la fotocopiadora mostrará en su pantalla un aviso señalando que se intenta fotocopiar un billete y que la acción no se completará por ser ilegal. De hecho, la medida es tan eficaz que si se intenta fotocopiar una hoja de papel en blanco con un eurión impreso, de un color adecuado, se producirá el mismo resultado que si se tratara de un billete de curso legal.

Transformación de mensajes: criptografía

El otro medio para lograr ocultar información, de modo que pueda ser recuperada por quien la emite o por su legítimo destinatario, pero que a la vez impida a un oponente acceder a ella, consiste en transformar el contenido de un mensaje

1. Véase http://www.rulesforuse.org/pub/index.php?currency=eur&lang=es

siguiendo determinadas reglas (en general, el mensaje original se suele denominar "mensaje en claro" o "texto claro"). Estas reglas modifican la información del mensaje, de modo que, aplicando las reglas inversas o adecuadas, será posible recuperar el mensaje original.

Este procedimiento de transformar un mensaje en claro en otro ininteligible, llamado *criptograma* o mensaje cifrado (texto cifrado), se conoce como *criptografía*, término que procede de la palabra griega *kryptos*, cuyo significado es "secreto", "oculto" o "disimulado". Así pues, el objetivo de la criptografía es permitir el intercambio de información haciendo el mensaje ilegible sin ocultar la existencia de dicho mensaje (Van Tilborg *et al.*, 2005).

El emisor del mensaje debe asegurarse de que las reglas que utiliza no serán fácilmente deducibles o supuestas por un atacante. En otro caso, el sistema utilizado no tendrá la validez que se espera de él y su uso perjudicará al emisor y al receptor.

De forma más general, el objetivo de la criptografía es garantizar que la información transmitida (o almacenada) posea las siguientes tres cualidades: *confidencialidad*, *integridad* y *autenticidad*. La confidencialidad consiste en lograr que la información permanezca secreta y solo sea conocida por quienes tienen autorización para ello. Por su parte, la integridad hace referencia a la necesidad de que la información no haya sido manipulada ni alterada desde su origen a su destino. Finalmente, la autenticidad obliga a que tanto el origen como la información transmitida sean auténticos, es decir, no se produzcan suplantaciones. Otras cualidades relacionadas con la información que considera la criptografía son su *disponibilidad* y *no repudio*.

El estudio para intentar alterar alguna de las cualidades de la información, anteriormente mencionadas, que persigue la criptografía se conoce como *criptoanálisis*. El objetivo de un criptoanalista es conocer la información original que el emisor transmite al receptor, para lo cual utilizará todos los medios a su alcance.

La unión de la criptografía y el criptoanálisis se conoce como *criptología*, si bien, por abuso de lenguaje, se suele hablar de criptografía para referirse a ambos conceptos.

A lo largo de la historia, las reglas criptográficas de transformación de los mensajes han ido modificándose, haciéndose cada vez más complejas y sofisticadas. Esta evolución ha corrido pareja a la de la tecnología. De hecho, hoy en día todos los métodos criptográficos hacen uso de los ordenadores, pues en caso contrario es muy probable que el sistema utilizado pueda ser vulnerado o roto y el contenido del mensaje conocido por un atacante o adversario.

Estas reglas de transformación suelen hacer uso de tres métodos diferentes: transposición, sustitución y cifrado.

El *método de transposición* consiste en barajar o recolocar las letras del mensaje, obteniendo uno o varios criptogramas (también llamados, en este caso, anagramas). Si el mensaje es corto, por ejemplo una única palabra, el método es inseguro porque solo hay unas pocas formas de combinar las letras de la palabra (como máximo el número de permutaciones de las letras). Sin embargo, el método se hará más complicado a medida que aumente la longitud del mensaje. No obstante, si se desea que el destinatario pueda recuperar la información, el anagrama no puede haberse generado al azar, sino siguiendo una regla, lo que facilita, a la vez, su análisis.

Para simplificar la notación y mientras no se diga lo contrario, los mensajes originales se escribirán en minúsculas, mientras que los transformados se escribirán en mayúsculas; además, las claves se escribirán en mayúsculas y en letra cursiva. Finalmente, no haremos uso ni de los espacios ni de los acentos para no dar pistas en los mensajes. Por ejemplo, el mensaje "criptografía" podría ser transformado, mediante transposición, en alguno de los siguientes, tengan sentido o no: "ATACIRFOGRIP, FATIGARPICOR, GRAFICOTRIPA, CRIPTAGARFIO". El número de posibles criptogramas que se pueden obtener a partir del mensaje original es de 59.875.200 (permutaciones de 12 letras de las que se repiten varias de ellas). Este número parece elevado,

pero con la ayuda de un ordenador es fácil obtener todos los criptogramas. Luego, bastará con buscar cuáles de ellos tienen sentido o, de forma más sencilla aún, buscar cuáles están en un diccionario. Sin embargo, si el mensaje fuera "consejosuperiordeinvestigacionescientificas", el número de posibles transposiciones a que da lugar este mensaje es mucho más elevado. De hecho, es el número de permutaciones de 43 letras entre las que hay varias repetidas. Tal número, en este caso, es:

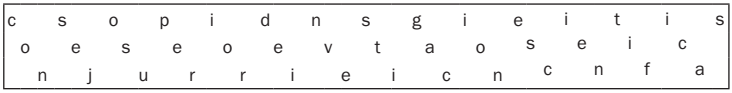
$$1.254.523.149.834.885.970.632.671.420.998.656.000.000 \approx 1{,}2 \cdot 10^{39}$$

Este número tiene 40 dígitos y para hacernos idea de cuán grande es podemos considerar las siguientes cifras: la población mundial puede estimarse en unos 7.000 millones de personas y la edad del universo, es decir, el tiempo transcurrido desde el Big Bang hasta ahora, es de unos 13.800 millones de años. Con estos datos, si una persona pudiera leer en un segundo una transposición de las posibles y si todas las personas del mundo trabajaran día y noche sin descanso, el número de transposiciones que habrían leído desde el principio del universo sería $7.000.000.000 \cdot 13.800.000.000 \cdot 365 \cdot 24 \cdot 60 \cdot 60$, es decir, habrían leído 3.046.377.600.000.000.000.000.000.000 transposiciones. Dividiendo el número de posibles transposiciones del mensaje anterior entre el número de transposiciones leídas por toda la población mundial desde el Big Bang, obtendríamos un valor de 411.808.158.600. En resumen, toda la población del universo debería repetir más de 411.000 millones de veces la lectura de una transposición por segundo durante toda la vida del universo para agotar toda la lista.

El destinatario legal podría, sin embargo, recuperar el mensaje original fácilmente a partir del transformado si supiera cómo se transformó el primero en el segundo, dado que bastaría con aplicar la transformación inversa.

Una de estas transformaciones podría ser una sencilla regla, fácil de recordar, como es la conocida como "regla del

riel" de tres filas (de forma análoga se podrían considerar rieles de dos filas, cuatro, etc.). Esta regla consiste en escribir el mensaje alternando sus letras en tres filas consecutivas y separadas, de la siguiente forma:

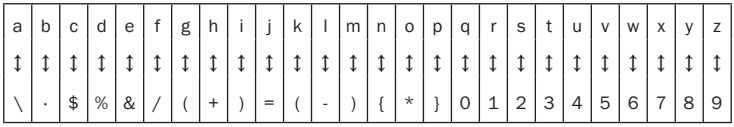


De este modo, el mensaje transformado sería en realidad el siguiente: "CSOPIDNSGIEITISOESEOEVTAOSEICNJURRIEICNCNFA".

Para recuperar el mensaje original bastaría con aplicar la regla inversa. Ahora bien, si el adversario supiera o tuviera motivos para pensar que la transposición anterior se ha logrado mediante algún sistema relacionado con la regla del riel, el número de posibles transformaciones que tendría que analizar sería mucho menor y sus posibilidades de éxito mucho mayores.

El "método de sustitución" cambia unas letras del mensaje por otras letras o por símbolos. De este modo, salvo que se conozca la equivalencia entre las primeras y los últimos, el mensaje original puede resultar muy difícil de recuperar. Por el contrario, si se dispone del diccionario de sustitución, este proceso será relativamente fácil.

Un sencillo ejemplo de método de sustitución puede ser el de seguir la regla dada por el siguiente diccionario:



De este modo, el mensaje "criptologia" se transforma en el siguiente: $1)}3*-*()\

Parece obvio que, salvo que se conozca el diccionario utilizado, no será fácil recuperar el mensaje original. Así pues, es importante que tanto el remitente del mensaje como su

destinatario tengan la misma copia del diccionario y que la guarden en lugar seguro para evitar su pérdida o robo.

Finalmente, el "método de cifrado" consiste en codificar las letras del mensaje de modo que se transformen en números y luego efectuar determinadas operaciones matemáticas con ellos. Parece claro que para recuperar el mensaje original se deberán realizar las operaciones en orden inverso (o las operaciones inversas a las originales) y luego descodificar los números obtenidos para transformarlos en letras y poder leer el mensaje.

Debe recordarse que la Real Academia Española (RAE, 2015) define, en su primera acepción, el verbo *cifrar* como "transcribir en guarismos, letras o símbolos, de acuerdo con una clave, un mensaje cuyo contenido se quiere ocultar". En este caso, la clave serán las operaciones realizadas y su orden.

Por otra parte, *codificar* es "transformar mediante las reglas de un código la formulación de un mensaje". Debe notarse que en este caso no se considera que esta transformación requiera ser secreta. De hecho, la mayoría de los códigos que empleamos no tiene como finalidad ser secreto, sino escribir la información de otro modo, con determinados objetivos, según el código empleado: código Morse (para ser enviado por telégrafo), Braille (para ser leído por invidentes), de circulación (para identificar mediante dibujos las pautas a seguir en el tráfico), de barras (para su lectura por medios electrónicos), etc.

Podría pensarse que el método de cifrado es una forma particular del método de sustitución, donde se cambian letras por números. Sin embargo, aquí los consideramos distintos porque existen métodos de sustitución que hacen exactamente eso: cambiar letras por números, pero en la cifra se va más lejos: primero se codifican las letras en números (no de forma secreta) y luego se realizan determinadas operaciones matemáticas que son el fundamento de la seguridad del método.

A modo de ejemplo, para cifrar el mensaje "codigo" se puede codificar primero dicha palabra siguiendo el Código

Estándar Estadounidense para el Intercambio de Información (*American Standard Code for Interchange of Information*, ASCII) y luego cifrar el número resultante llevando a cabo diferentes operaciones. El ASCII extendido codifica letras, números y caracteres mediante 8 bits (lo que da lugar a $2^8 = 256$ caracteres codificables). Algunos de estos caracteres y su codificación en binario y decimal se presentan en la tabla 1.

Tabla 1
Algunos caracteres del código ASCII.

CARÁCTER	ESCAPE	1	ñ	Ñ	©	Ë
Binario	00011011	00110001	10100100	10100101	10111000	11010011
Decimal	27	49	164	165	184	211

Así, la letra *A* se codifica, en binario, como 01000001 (el 65 en decimal); la *B* como 01000010 (el 66); la *a* mediante 01100001 (el 97), etc. Es decir, el ASCII asigna a las letras *a*, *b*, *c*…, *z*, los números 97, 98, 99…, 122, por lo que la codificación de cada una de las letras del mensaje anterior da lugar al siguiente grupo de números: "99 111 100 105 103 111".

Si ahora consideramos que el cifrado de cada número consiste en multiplicarlo por 23, luego dividir el producto entre 256 y finalmente considerar como resultado el resto de esa división, tenemos que:

$99 \cdot 23 = 2277,$
$2277 = 256 \cdot 8 + \underline{229}.$

Por lo que la letra *c*, que se codifica como 99, pasa a ser cifrada como 229.

Repitiendo el proceso para todos los demás números, se tiene que el mensaje cifrado es: "229 249 252 111 65 249".

Para descifrar el criptograma y recuperar el mensaje original, se debe multiplicar cada uno de los números que forman el criptograma por 167, luego dividir el producto entre

256 y considerar como resultado el resto de esa división (más adelante, en el apartado "Algunos conceptos matemáticos" del capítulo 3, se explicará por qué este proceso invierte las operaciones anteriores). A modo de ejemplo, para el 229 se obtiene lo siguiente:

$229 \cdot 167 = 38243,$
$38243 = 256 \cdot 149 + \underline{99}.$

Ahora, se descodifica el 99 obtenido siguiendo el ASCII y resulta la letra *c*. Para el resto de los números se procede de modo análogo.

CAPÍTULO 1

Criptografía clásica

Como ya hemos señalado en la Introducción, la criptografía tiene como objetivo permitir que dos personas puedan intercambiarse información de forma confidencial y segura mediante un canal inseguro. Se trata, en definitiva, de que solo los destinatarios autorizados puedan recuperar la información oculta aunque esta pueda ser interceptada por enemigos o atacantes.

Para llevar a cabo este proceso, el mensaje original o texto claro se transforma mediante alguno de los métodos ya mencionados (transposición, sustitución o cifrado) en el texto oculto, texto cifrado o criptograma, haciendo uso de unas reglas o claves.

En este capítulo daremos un pequeño repaso a algunos de los métodos que se han utilizado a lo largo de los años, desde la época clásica griega hasta finales del siglo XIX, para ocultar a los adversarios la información que se desea transmitir.

'Escítala lacedemonia'

En primer lugar, vamos a recordar las formas en las que los clásicos mantenían en secreto sus mensajes. Debe tenerse en cuenta que, en general, se trataba de mensajes con contenido

militar y el objetivo era comunicarse con las propias tropas de modo rápido, evitando que el mensaje fuera conocido por el enemigo si el mensajero era capturado.

Uno de los primeros métodos criptográficos documentados es la llamada *escítala lacedemonia*, por ser utilizada por los espartanos. De hecho, fue el primer sistema criptográfico militar de la historia (siglo V a.C.) del que se tiene noticia, y fue utilizado por Alejandro Magno (356-323 a.C.).

La *escítala* era una vara de madera en la que se enrollaba, a lo largo y en espiral, una cinta de pergamino, de modo que el mensaje que se deseaba enviar se escribía de arriba abajo de la vara (no a lo largo del pergamino). Posteriormente, el pergamino se desenrollaba y se enviaba a su destinatario. Con este sistema, las letras que forman el mensaje original quedan distribuidas en otro orden, dependiendo del diámetro de la vara.

Para recuperar el mensaje original, basta con repetir el proceso anterior utilizando una vara con el mismo diámetro. Este diámetro podría considerarse como la clave, dado que si la vara empleada por el receptor no tiene el mismo diámetro que la originalmente utilizada el mensaje que se desea transmitir no podría leerse verticalmente.

Hoy en día se podría emplear el mismo método de transposición de un modo más discreto: bastaría con utilizar un lápiz o un cilindro de cartón como los que se emplean en los rollos de papel de cocina (a modo de vara) y una tira estrecha de papel.

Haciendo un poco de abstracción también se podría pensar que la *escítala* define una tabla de tantas columnas como letras se pueden escribir en una vuelta del papel a la vara y tantas filas como letras quepan a lo largo de la misma. A modo de ejemplo, si el número de columnas fuera de cinco y el de filas de seis, el mensaje "loquenosabemosdelacriptografia" podría escribirse por columnas como se indica a continuación:

l	o	o	c	g
o	s	s	r	r
q	a	d	i	a
u	b	e	p	f
e	e	l	t	i
n	m	a	o	a

Posteriormente, se leería por filas, de modo que el criptograma resultante sería: "LOOCGOSSRRQADIAUBEPFEELTINMAOA".

Como se puede apreciar, el criptograma contiene exactamente las mismas letras que el mensaje original, pero en un orden distinto, que depende del diámetro de la vara (en este caso, del número de columnas). Así pues, la *escítala* es un método criptográfico de transposición.

'Atbash'

El método criptográfico clásico de sustitución por antonomasia consiste en considerar un alfabeto de n letras y luego aplicar la siguiente regla a modo de clave: "La letra que ocupa la posición i en el alfabeto se sustituye por la que ocupa la posición $n-i+1$ del mismo alfabeto". Con esta sencilla regla, si consideramos el alfabeto de $n = 26$ letras (sin la ñ), la letra *a* (posición 1) se cambiaría por la *z* (posición 26); la *b* por la *y*; la *c* por la *x*, y así sucesivamente (y viceversa). Es decir, tendríamos el siguiente diccionario de sustitución:

a	b	c	d	e	f	g	h	i	j	k	l	m
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
z	y	x	w	v	u	t	s	r	q	p	o	n

Parece claro que si no se conociera la regla de sustitución, recuperar el mensaje original "antiguotestamento" a partir de su criptograma "ZMGRTFLGVHGZNVMGL" no sería tan fácil. Este método de sustitución se emplea en

diferentes ocasiones en el Antiguo Testamento y es una forma hebrea muy antigua de ocultamiento de información conocida como *atbash*.

La propia palabra *atbash* da pistas sobre la sustitución que se lleva a cabo. En efecto, si se considera el alfabeto hebreo, su primera letra es א (*aleph*), mientras que la última es ת (*taf*); la segunda letra es ב (*bet*) y la penúltima es ש (*shin*), lo que da lugar al término anterior. Una forma análoga de llamar a este tipo de sustitución utilizando exclusivamente nuestro alfabeto sería *azby*.

Cifrado de Julio César

Parece ser que Julio César (100-44 a.C.) utilizó numerosos métodos criptográficos. Sin embargo, hasta nosotros solo ha llegado uno de ellos, el método que consiste en sustituir cada letra del mensaje original por la tercera letra que le sigue en el alfabeto, es decir, la letra *a* se sustituye por la *D*, la *b* por la *E*, la *c* por la *F*, y así sucesivamente. En este caso, podríamos decir que la clave es el número tres (o la letra *D*) porque es el valor que indica la posición de la letra posterior que sustituye a cada letra del mensaje original. El diccionario de sustitución de César sería el siguiente:

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
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
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

De este modo, se puede entender que el alfabeto original se sustituye por otro alfabeto que resulta de reordenar el primero. A veces este tipo de sustitución se denomina "monoalfabética".

Si César hubiera querido ocultar su famosa frase "veni vidi vici" mediante el método de sustitución anterior, habría obtenido: "YHQL YLGL YLFL".

Este ejemplo pone de manifiesto, de forma visual, una de las debilidades de algunos sistemas criptográficos y que estudiaremos con más detalle posteriormente (apartado "Criptoanálisis de los sistemas clásicos" en este mismo capítulo): cada letra del mensaje original se sustituye siempre por la misma letra en el criptograma (lo mismo sucedía con el *atbash*), lo que hace que la frecuencia de las letras de un mensaje escrito en determinado idioma se mantenga exactamente en los criptogramas que se obtienen.

Si alguien nos hubiera dicho que el criptograma anterior lo llevaba un mensajero de César, probablemente no hubiéramos tardado mucho en saber qué mensaje ocultaba. En efecto, basta tener en cuenta que las tres palabras empiezan por la misma letra, que las tres terminan por la misma letra y que estas son iguales a las segundas letras de las dos últimas palabras, etc. Todos estos datos nos llevarían a deducir el contenido del mensaje original.

Hoy en día, por extensión y por abuso de lenguaje, cualquier sistema criptográfico que siga un método de sustitución con una regla o clave similar a la empleada por Julio César se conoce como "cifrado de César". Por ejemplo, pertenecerían a este grupo sistemas cuyas reglas fueran "sustituir cada letra por la *n*-ésima siguiente" siendo *n* cualquier número menor que el tamaño del alfabeto; "sustituir cada letra que ocupa una posición par por la *n*-ésima siguiente y cada letra que ocupa una posición impar por la *n*-ésima anterior"; "sustituir la primera letra por la siguiente, la segunda por la que está dos posiciones después, la tercera por la que ocupa tres posiciones más adelante", etc.

Cifrado de Polibio

Otro sistema de sustitución, que en este caso utiliza números, fue descrito alrededor del 150 a.C. por el historiador griego Polibio (200-118 a.C.) en sus *Historias* (Polibio,

1997) y atribuido a dos de sus contemporáneos: Cleoxeno y Democleto. En este caso, se trata de sustituir cada letra del alfabeto por un par de dígitos. La forma más sencilla de este sistema consiste en formar una tabla cuadrada de 25 celdas, escribir en cada una de ellas una letra (solo se pueden utilizar $25 = 5^2$ letras dado que la tabla es cuadrada) y numerar las filas y columnas de la tabla con números del 1 al 5 como se muestra en la tabla 2.

Tabla 2
Cifrado de Polibio.

	1	**2**	**3**	**4**	**5**
1	a	b	c	d	e
2	f	g	h	j	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

Este tipo de tabla tiene la ventaja de que solo utiliza cinco símbolos diferentes: 1, 2, 3, 4 y 5, y dado que sustituye las letras por números, es posible llevar a cabo operaciones aritméticas con los mismos. A modo de ejemplo, el mensaje "polibio" proporcionaría el siguiente criptograma (hemos considerado que, como solo se pueden utilizar 25 letras, las letras *i* y *j* se sustituyen ambas por la *j*): "35 34 31 24 12 24 34". Por otra parte, la tabla de Polibio puede generalizarse de modo similar a como se hizo con el cifrado de César: basta pensar en una palabra como clave y colocar sus letras como las primeras de la tabla y, a continuación, escribir las restantes letras del alfabeto en su orden lexicográfico. Por ejemplo, con la clave *TABLERO* se tendría la tabla 3, de modo que el criptograma correspondiente al mensaje anterior pasaría a ser: "42 22 14 33 13 33 22".

TABLA 3
Cifrado de Polibio con la clave *tablero*.

	1	2	3	4	5
1	t	a	b	l	e
2	r	o	c	d	f
3	g	h	j	k	m
4	n	p	q	s	u
5	v	w	x	y	z

Es importante destacar que, sin importar la clave que se utilice, las letras repetidas en el mensaje original, también en este caso, se transforman en el mismo par de dígitos en el criptograma. Dicho de otra forma, la frecuencia de las letras del mensaje original se mantiene en el criptograma como pares de dígitos.

Criptoanálisis de los sistemas clásicos

Ya hemos mencionado que todo sistema criptográfico es susceptible de ser analizado con vistas a recuperar el mensaje oculto en el criptograma. Hoy en día, la mayor parte de los sistemas clásicos pueden ser criptoanalizados y vulnerados de forma sencilla, es decir, es posible recuperar la clave que se utilizó en el proceso de cifrado y obtener el mensaje original.

Hay que tener en cuenta que actualmente tenemos muchos más conocimientos de cómo funcionan los sistemas de cifrado y que poseemos, además, numerosas herramientas que nos permiten analizar con detalle los sistemas clásicos. Piénsese que nuestras ventajas son enormes comparadas con las de los adversarios que intentaban descubrir los mensajes ocultos en los criptogramas clásicos. Para empezar, baste decir que, entre otras cosas, había que saber leer y escribir, lo que no era una habilidad demasiado extendida.

Existen, fundamentalmente, tres métodos de ataque contra los sistemas clásicos: por fuerza bruta (o búsqueda exhaustiva de claves), por máxima verosimilitud y estadísticos.

En los cifrados por sustitución de tipo César, por ejemplo, una primera forma de intentar hallar la clave empleada consiste en probar todas y cada una de las posibles claves, es decir, proceder por *fuerza bruta*. Si el número de las mismas no es excesivamente elevado, puede que, con lápiz y papel, en solo unas horas consigamos nuestro objetivo (en segundos si hacemos uso de ordenadores). De hecho, si se sabe que el criptograma "ZUJUY RUY IGSOTUY RRKBGT G XUSG" corresponde a un mensaje transformado por un cifrador de tipo César, dado que solo hay 25 posibles claves (tantas como letras hay en el alfabeto menos una) bastaría con ir probando con cada una de ellas hasta obtener un mensaje con sentido.

Por ejemplo, podríamos probar si la clave que se empleó fue el 1 (que correspondería a sustituir la *a* por la *b*, etc.), luego el 2 (sustituir la *a* por la *c*, etc.) y así sucesivamente hasta llegar a descubrir el mensaje original. El proceso mencionado se muestra en la tabla 4.

Tabla 4
Ataque por fuerza bruta a un cifrado César.

TEXTO	CLAVE
ZUJUY RUY IGSOTUY RRKBGT G XUSG	
avkvz svz jhtpuvz sslchu h yvth	1
bwlwa twa kiuqvwa ttmdiv i zwui	2
cxmxb uxb ljvrwxb uunejw j axvj	3
...	...
todos los caminos llevan a roma	6

En este caso, romper el sistema es sencillo, dado que sabemos que cada letra se sustituye por otra que ocupa una determinada posición hacia la derecha. Sin embargo, si solo

se supiera que el alfabeto tiene 26 letras y que se ha usado el método de sustitución, sin más datos, determinar la clave es una tarea mucho más complicada. En efecto, dado que la sustitución de una letra podría ser cualquiera de las 26 del alfabeto (incluso alguna podría sustituirse por ella misma), la de otra letra sería cualquiera de las 25 restantes, y así sucesivamente, es decir, el número de posibles claves sería el mismo que el número de permutaciones que se pueden conseguir con 26 letras, esto es,

$$26! = 26 \cdot 25 \cdots 3 \cdot 2 \cdot 1 = 403.291.461.126.605.635.584.000.000 \approx 4 \cdot 10^{26}$$

que es un número excesivamente grande como para plantearse una búsqueda exhaustiva de la clave.

No obstante, en determinadas ocasiones, hacer uso de posible información contenida en el mensaje original, saber de qué se está hablando o utilizar la estructura del idioma en que está escrito el mensaje puede ayudar a recuperar el mensaje original. Se trata de llevar a cabo un ataque por *máxima verosimilitud*.

Volviendo al ejemplo del criptograma anterior, donde hemos mantenido los espacios originales del mensaje por razones didácticas, podemos razonar de la siguiente manera: en primer lugar, se puede observar que la quinta palabra está formada por una única letra. Esto da una gran pista porque podríamos pensar que originalmente era la conjunción *y*, con lo que la *G* se habría sustituido por la *y*; es decir, la clave sería 18. Pero si así fuera, la última letra de la primera palabra (o de la segunda o la tercera), *Y*, procedería de una *q*; y en español no hay palabras que terminen en *q*. Así pues, la clave no es 18.

Podemos pensar entonces que la *G* se ha sustituido por una *e*, que es otra conjunción en español, lo que supondría que la clave es 24 (o –2, si tomamos como criterio que el signo – significa retroceder). Ahora bien, dado que la *e* solo se emplea como conjunción si la siguiente palabra empieza por *i*, resulta que tampoco es esta la clave porque la siguiente palabra comenzaría por *v* (proviene de una *X*).

De nuevo podríamos pensar que la *G* ha sido sustituida por la conjunción *o*, siendo la clave 8. Pero tampoco es así porque entonces la última letra de la primera palabra, *Y*, procedería de una *g* y no hay palabras en español que terminen por *g*.

Otra opción sería la de suponer que la *G* se ha sustituido por la preposición *a*, con lo que la clave sería 20 (o –6). En este caso no obtenemos razones para desechar esta clave, dado que proporciona el mensaje original: "todos los caminos llevan a roma".

Parece claro que podría haberse utilizado un criterio o metodología diferente para intentar romper el sistema de sustitución anterior.

Finalmente, una tercera forma de atacar los métodos de sustitución es utilizar herramientas *estadísticas*; en particular, usar el hecho de que la frecuencia de las letras del mensaje original se mantiene en las letras o símbolos que los sustituyen en el criptograma. Si a esto le añadimos que conocemos la frecuencia de repetición de las letras en un idioma cualquiera (cosa fácil hoy en día con el uso de ordenadores), resulta que podemos intentar utilizar tal conocimiento estadístico para recuperar un mensaje.

Parece ser que la primera explicación del uso del análisis de frecuencias de las letras como método de criptoanálisis se debe al filósofo árabe Al-Kindi (o Abū Yūsuf Ya´qūb ibn Isḥāq al-Kindī, 801-873), quien la publicó en su obra *Un manuscrito para el descifrado de mensajes criptográficos*.

En español el orden de las letras generalmente aceptado, según su frecuencia, es: "e a o s r n i l d c t u p m y q g v h f b j z k w x". No obstante, este conocimiento no es suficiente, dado que no siempre la distribución de las letras en el texto original corresponderá exactamente a este orden y, en ocasiones, se debe recurrir a mayor información como saber el orden de la frecuencia de pares de letras: "es en el de la os ue ar ra re on er as st al ad" y tríos de letras más frecuentes: "que est ara ado aqu cio del nte ede osa per nei ist sde".

En todo caso, las letras y grupos de letras anteriores pueden dar una idea de cuáles son más fáciles de encontrar

en un texto, pero su frecuencia depende del tipo de texto que se haya considerado de referencia para hacer el cómputo.

Posiblemente uno de los ejemplos más claros y divertidos que ha ofrecido la literatura universal sobre la forma de realizar el criptoanálisis de un texto haciendo uso de esta herramienta es el que se presenta en la novela *El escarabajo de oro* (Poe, 2007) de Edgar Allan Poe (1809-1849). En este caso, el protagonista se encuentra con un documento críptico y sospecha que contiene las instrucciones para llegar al tesoro escondido de un pirata. El documento (escrito en inglés) es:

```
5 3 ‡ † 3 0 5 ) ) 6 * 4 8 2 6 ) 4 ‡ . ) ‡ ) ; 8 0 6
* ; 4 8 ‡ 8 ¶ 6 0 ) ) 8 5 ; 1 ‡ ( ; : ‡ * 8 † 8 3 (
8 8 ) 5 * † ; 4 6 ( ; 8 8 * 9 6 * ? ; 8 ) ‡ ( ; 4 8
5 ) ; 5 * † 2 : * ‡ ( ; 4 9 5 6 * 2 ( 5 * - 4 ) 8 ¶
8 * ; 4 0 6 9 2 8 5 ) ; ) 6 † 8 ) 4 ‡ ‡ ; 1 ( ‡ 9 4
8 0 1 ; 8 : 8 ‡ 1 ; 4 8 † 8 5 ; 4 ) 4 8 5 † 5 2 8 8
0 6 * 8 1 ( ‡ 9 ; 4 8 ; ( 8 8 ; 4 ( ‡ ? 3 4 ; 4 8 )
4 ‡ ; 1 6 1 ; : 1 8 8 ; ‡ ? ;
```

A lo largo de la novela se narran las peripecias, razonamientos y suposiciones de Legrand, el protagonista, para encontrar el contenido original del mensaje.

Otro ejemplo es la aventura de Sherlock Holmes titulada *El misterio de los bailarines* o, en ocasiones, *Los hombres danzantes* (Doyle, 2013), de sir Arthur Conan Doyle (1859-1930). En este caso, el señor Cubitt recurre a Sherlock Holmes porque su esposa está aterrada al haber recibido unos papeles en los que aparecen unos muñecos que parecen danzar, similares a los que se muestran en la figura 2.

Holmes deduce que lo que parecen muñecos infantiles pintados al azar no son sino letras que ocultan un mensaje. El proceso deductivo de Holmes es similar al empleado en *El escarabajo de oro*, con algunas variaciones, como el hecho de deducir que si un bailarín lleva una bandera es señal de que marca el final de una palabra o frase.

FIGURA 2
Mensaje mediante bailarines.

FUENTE: ELABORACIÓN PROPIA.

Una forma de evitar este tipo de ataques estadísticos consiste en recurrir al uso de unos elementos llamados homófonos y nulos.

Un *homófono* es una palabra que suena igual que otra, pero que tiene diferente significado (uso y huso, tuvo y tubo son ejemplos de pares de homófonos). Se trata, por tanto, de utilizar diferentes símbolos que sustituyan a una misma letra, de modo que la frecuencia de su repetición en el mensaje original quede dispersa. Este sistema hace que el alfabeto de cifrado sea mayor que el de partida, tanto más cuanto mayor sea el número de homófonos para una misma letra.

Por el contrario, los *nulos* son símbolos que no tienen significado alguno y que se incluyen en el criptograma para complicar el descifrado no autorizado de mensajes.

A lo largo de la historia se ha hecho uso de diferentes métodos para insertar homófonos y nulos en los sistemas de sustitución. Uno de ellos, relativamente moderno, fue el empleado en el Laboratorio Nacional Los Álamos (Estados Unidos), donde se llevaron a cabo las investigaciones para el desarrollo de la bomba atómica durante la Segunda Guerra Mundial. En este complejo se utilizaban homófonos y nulos para las conversaciones telefónicas con el exterior, como los mostrados en la tabla 5.

Se puede observar que el cuadro contiene 100 celdas capaces de permitir varias combinaciones de pares de números (sustituciones) para una misma letra (homófonos), así como utilizar otros pares para no representar nada (nulos). De este modo, la letra *e* puede ser sustituida por los siguientes pares de números: 22, 25, 31, 49, etc., mientras que los pares de dígitos 14, 18, 27, 46, etc., corresponden a nulos.

TABLA 5

Homófonos y nulos utilizados en Los Álamos.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**
1	i	p	i		o	u	o		p	n
2	w	e	u	t	e	k		l	o	
3	e	u	g	n	b	t	n		s	t
4	t	a	z	m	d		i	o	e	
5	s	v	t	j		e		y		h
6	n	a	o	l	n	s	u	g	o	e
7		c	b	a	f	r	s		i	r
8	i	c	w	y	r	u	a	m		n
9	m	v	t		h	p	d	i	x	q
0	l	s	r	e	t	d	e	a	h	e

A modo de ejemplo, el mensaje "losalamos" podría dar lugar a cualquiera de los siguientes dos criptogramas: "28 71 15 39 62 64 38 08 88 69 66 14" y "01 63 77 94 74 28 87 44 63 51".

Obsérvese que, en este caso, el mensaje original contiene dos letras *l*, dos letras *o*, dos letras *s*, dos letras *a* y una *m*. Sin embargo, ninguno de los dos criptogramas contiene pares de dígitos repetidos y, además, las longitudes de los mismos son diferentes. Este tipo de estrategia permite engañar, de alguna manera, a la estadística de las letras del mensaje original.

En cualquier caso, lo que sí se sigue verificando, se haga uso de homófonos o no, es que el proceso de recuperar el mensaje original es único, es decir, no importa cuál sea el criptograma del que se parta, el mensaje original recuperado siempre será el mismo.

No obstante, no todo está perdido para los adversarios que deseen recuperar un mensaje a partir de un criptograma. Existe una debilidad en el uso de homófonos y nulos y esta se presenta cuando un mismo texto contiene varias veces la misma palabra. En este caso, las sustituciones de dicha palabra no tienen muchas formas diferentes de aparecer en el criptograma y ello puede ayudar a eliminar el factor de los homófonos y nulos y volver a utilizar las frecuencias de las letras como método de ataque.

Cifradores de rotación

Con el paso del tiempo, los sistemas de sustitución fueron mejorándose para dificultar el trabajo de los adversarios que intentaban recuperar los mensajes originales a partir de los criptogramas.

Uno de estos avances se debió a Leon Battista Alberti (1404-1472) mediante los que se han dado en llamar los "cifradores de rotación" o "discos cifradores" (nótese, de nuevo, el abuso de lenguaje, dado que en realidad son sistemas de sustitución). La idea de Alberti, publicada en su obra *Modus scribendi in ziferas*, fue la de utilizar no solo un alfabeto para sustituir unas letras por otras, sino más de un alfabeto, es decir, llevar a cabo una sustitución según una regla y luego, después de determinado número de sustituciones, cambiar de regla.

Ya comentamos al hablar del cifrado de César que se trataba de una sustitución monoalfabética porque cada letra original se sustituía por otra letra de un alfabeto reordenado. Ahora nos encontramos con una nueva idea que hace uso de varios alfabetos reordenados de diferente manera, de ahí que se llamen "sustituciones polialfabéticas".

Una forma sencilla de entender la idea de Alberti consiste en utilizar, por ejemplo, dos alfabetos de sustitución, de modo que a la hora de llevar a cabo las sustituciones de las letras del mensaje original se alterne el uso de ambos alfabetos. Escribiendo ordenadamente el alfabeto ordinario y los dos de sustitución tendríamos:

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

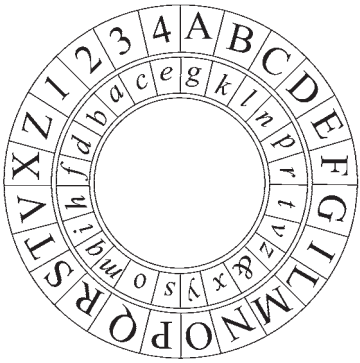
De este modo, el criptograma correspondiente al nombre "leonbattistaalberti" sería el resultado de sustituir la primera letra mediante el primer alfabeto, la segunda mediante el segundo, la tercera mediante el primero, etc. Es decir, las

letras que ocupan posiciones impares se sustituyen por sus correspondientes del primer alfabeto y las que ocupan posiciones pares por las del segundo alfabeto. Así, el criptograma resultante sería: "CILAMVUSZXUVJJMIOSZ".

Rápidamente se aprecia que una misma letra del mensaje original no siempre se transforma en la misma letra en el mensaje cifrado.

Una reproducción del cifrador de Alberti se presenta en la figura 3. El cifrador consiste en dos círculos concéntricos que giran de forma independiente, de modo que, una vez que se han fijado sus posiciones relativas, las letras del círculo interior se sustituyen por las correspondientes del círculo exterior. Después de sustituir determinado número de letras, se puede modificar el giro de los círculos y seguir, de este modo, con un nuevo alfabeto de sustitución (Sgarro, 1990).

Figura 3
Disco cifrador de Alberti.



Fuente: Elaboración propia.

Ya hemos mencionado a Della Porta a propósito de las tintas invisibles, pero también participó en el desarrollo de los cifradores polialfabéticos (Della Porta, 1996), proponiendo un cifrador de rotación diferente, como se puede ver en la figura 4.

Figura 4
Disco cifrador de Della Porta.

Fuente: Extraído de la obra de Giovanni Battista della Porta.

Además de las propuestas de Alberti y Della Porta, mencionadas más arriba, hicieron falta los desarrollos del abad alemán Johannes Trithemius (1462-1516) y, sobre todo, del diplomático francés Blaise de Vigenère (1523-1596) para que los cifradores polialfabéticos se convirtieran en una realidad, como veremos más adelante.

El sistema de los discos cifradores se ha utilizado hasta no hace mucho, como puede comprobarse con la reproducción (por la Agencia Nacional de Seguridad estadounidense, NSA, National Security Agency) del disco cifrador de la Confederación empleado en la guerra civil americana, mostrado en la figura 5.

FIGURA 5
Reproducción de la NSA de un disco cifrador confederado.

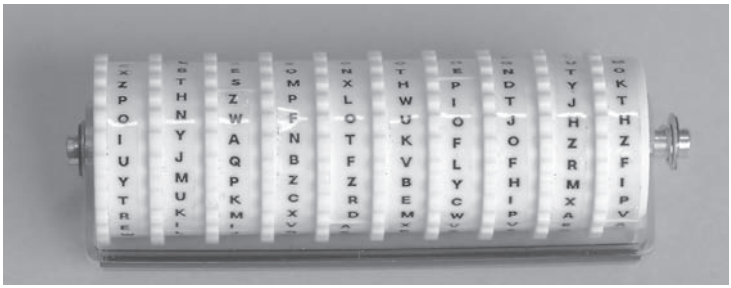
FUENTE: FOTOGRAFÍA CEDIDA POR EL CCN.

Un sistema más complicado, pero basado en la misma idea, fue el cifrador cuyo diseño se atribuyó al tercer presidente de los Estados Unidos, Thomas Jefferson (1743-1826), aunque no ha sido probado. Este cifrador es un cilindro que consta de 36 discos de madera, con las 26 letras del alfabeto cada uno, ordenadas de manera diferente y que giran alrededor de un eje (en la fotografía de la figura 6 se muestra una reproducción en plástico). El mensaje en claro se formaba moviendo los discos de madera hasta que este aparecía en una de las líneas del cilindro, transmitiéndose como criptograma cualquiera otra de las líneas con la formación de los discos fijada. Para recuperar el mensaje original bastaba con formar la línea recibida moviendo los discos y leer entre las restantes líneas la que tuviera sentido. Es evidente que el receptor debía conocer el orden en el que se debían colocar los discos en el cilindro. Un adversario, por el contrario, además de conocer la distribución de las letras en los 36 cilindros, tendría que probar las 36! maneras de colocar los discos en el cilindro:

$$36! = 371.993.326.789.901.217.467.999.448.150.835.200.000.000 \approx 3 \cdot 10^{41}$$

Figura 6

Reproducción del cifrador atribuido a Thomas Jefferson.



Fuente: Reproducción cedida para ser fotografiada para este libro por el CCN. Fotografía realizada por J. Negrillo.

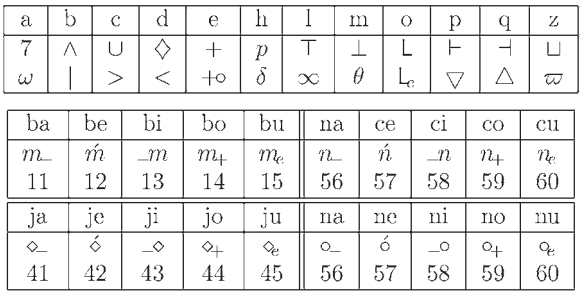
Nomenclátor de Felipe II

El rey Felipe II (1527-1598), como todos los monarcas europeos de la época, también utilizaba métodos de cifra para enviar información secreta a sus tropas y a sus aliados. Sin embargo, en lugar de emplear sistemas polialfabéticos como ya se estaba haciendo en Europa, para cifrar sus mensajes hacía uso de un nomenclátor, es decir, un catálogo (RAE, 2015) de nombres propios o de voces técnicas que en este caso se utilizaba para ocultar la información al enemigo.

De hecho, los criptógrafos de Felipe II utilizaban una cifra monoalfabética de sustitución extendida, consistente en un abecedario, un silabario y un diccionario. El abecedario (parte del mismo se puede ver en la imagen superior de la figura 7) consistía en una tabla de sustitución para las letras, con homófonos, mientras que el silabario (imagen inferior de la figura 7) permitía cifrar sílabas. Por su parte, el diccionario no era más que una recopilación de abreviaturas para los términos más utilizados en los mensajes. Así, la palabra "alemanes" se abreviaba con las letras "eb"; mientras que "Cartagena" era "bra", etc.

FIGURA 7

Reproducción parcial del abecedario y del silabario de Felipe II.



FUENTE: ELABORACIÓN PROPIA.

No obstante, a pesar de la complejidad aparente de este nomenclátor, muchos de los mensajes cifrados de Felipe II fueron criptoanalizados por los franceses, en especial por el matemático François Viète (1540-1603). Cuando los españoles descubrieron que sus mensajes cifrados eran desvelados sistemáticamente por Viète, Felipe II pidió al Vaticano que el matemático francés fuera juzgado por sus actos diabólicos, dado que era un "enemigo jurado confabulado con el diablo". Sin embargo, el papa no hizo caso de la petición española, posiblemente porque los criptoanalistas del Vaticano también eran capaces de leer los mismos mensajes secretos.

Cifra indescifrable

La verdadera revolución en los cifradores polialfabéticos la llevó a cabo Blaise de Vigenère con la publicación en 1586 de su obra *Traicté des Chiffres*. En ella propone el uso de 26 alfabetos distintos para cifrar un mensaje, de modo que el primer alfabeto es el alfabeto original en el que la primera letra pasa a ser la última, esto es, comienza por B C D y termina con Y Z A (sería el equivalente a un cifrado de César con clave 1). El segundo alfabeto repite la operación anterior con el primer

alfabeto, es decir, es: C D E… Z A B, y así sucesivamente hasta volver al alfabeto inicial. De modo esquemático, el cuadro de Vigenère se presenta en la tabla 6.

TABLA 6
Cuadro de Vigenère.

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**…**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**z**
1	B	C	D	E	F	G	H	I	…	T	U	V	W	X	Y	Z	A
2	C	D	E	F	G	H	I	J	…	U	V	W	X	Y	Z	A	B
3	D	E	F	G	H	I	J	K	…	V	W	X	Y	Z	A	B	C
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
24	Y	Z	A	B	C	D	E	F	…	Q	R	S	T	U	V	W	X
25	Z	A	B	C	D	E	F	G	…	R	S	T	U	V	W	X	Y
26	A	B	C	D	E	F	G	H	…	S	T	U	V	W	X	Y	Z

Una vez construido el cuadro anterior, para cifrar un mensaje se debe elegir una clave, que es la que decidirá qué alfabetos de los 26 existentes van a utilizarse para cifrar el texto original. A modo de ejemplo: si se desea cifrar el mensaje "cifraindescifrable" con la clave *BLAISE* se procede como sigue: la clave se repite debajo del mensaje a cifrar tantas veces como sea preciso, de modo que se consigan tantas letras como tiene el mensaje original. Cada letra de la clave indica qué alfabeto se emplea para cifrar la letra del mensaje que toque en cada momento. Esquemáticamente, sería así:

Mensaje	c	i	f	r	a	i	n	d	e	s	c	i	f	r	a	b	l	e
Clave	*B*	*L*	*A*	*I*	*S*	*E*	*B*	*L*	*A*	*I*	*S*	*E*	*B*	*L*	*A*	*I*	*S*	*E*
Criptograma	D	T	F	Z	S	M	O	O	E	S	U	M	G	C	A	J	D	I

A la vista del método anterior, queda claro que cuanto más larga sea la clave, mayor dificultad presenta el método de Vigenère para poder romperse. Además, es necesario que el destinatario conozca dicha clave para poder llevar a cabo el proceso inverso.

El método de Vigenère parecía realmente indescifrable; sin embargo, no fue así. Friedrich Wilhelm Kasiski (1805-1881), mayor retirado del ejército prusiano, fue el primero en publicar en 1863 un método general *(La escritura secreta y el arte del desciframiento* o *Die Geheimschriften und die Dechijfrirkunst)* para descifrar este método (Fúster *et al.*, 2004: 23).

Brevemente, el método de Kasiski consiste en calcular, en primer lugar, la longitud de la clave y, posteriormente, la clave. Para ello, se buscan palabras repetidas en el texto cifrado. Tales palabras corresponderán, muy probablemente, no solo a palabras repetidas en el texto original, sino que, además, la posición de la clave coincidirá para tales palabras. Este hecho significa que la distancia entre palabras repetidas es un múltiplo de la longitud de la clave, por lo que es casi seguro que el máximo común divisor de tales distancias sea la longitud de la clave.

Una vez que se conoce el número de letras de la clave, se trata de determinar la propia clave, para lo cual basta con dividir el texto en bloques con el tamaño de la clave y considerar que el cifrado es un cifrado tipo César.

Parece ser que el matemático británico Charles Babbage (1791-1871) llegó antes (en 1854) que Kasiski a los mismos resultados que este, pero no fueron publicados (Singh, 2000: 73).

Cifrado Playfair

El escocés lord Lyon Playfair (1818-1898) popularizó el sistema de sustitución que ahora lleva el nombre de cifrado de Playfiar, pero que fue inventado por su amigo el inglés sir Charles Wheatstone (1802-1875). El sistema recuerda al cifrado de Polibio, dado que las letras se disponen en un cuadrado de 5×5 celdas (en este caso las letras i y j también se consideran como la misma letra, dado que solo se pueden utilizar 25). La clave a utilizar da lugar a la distribución de las letras en el cuadrado. Así, la clave puede estar

formada por solo unas pocas letras del cuadrado o por todas ellas, y estas pueden comenzar al inicio del cuadrado o desde cualquier posición acordada por los usuarios que vayan a utilizar este sistema de sustitución. Si la clave establecida (que conviene sea fácil de recordar por emisor y receptor) fuera la frase *EL CIFRADO DE PLAYFAIR FUE INVENTADO POR WHEATSTONE* dado que hay varias letras repetidas, la clave se reduciría a la siguiente: *ELCJFRADOPYUNVTWHS*. Si se hubiera acordado, además, que la posición inicial de la clave fuera el centro del cuadrado, esta comenzaría a escribirse en tal posición, completándose el cuadrado con las letras del abecedario aún no empleadas y ordenadas lexicográficamente:

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

Una vez establecida la clave, se procede a transformar el mensaje, dividiéndolo en grupos de dos letras. Si el número de letras fuera impar, se añade una letra al final, por ejemplo una *x*. Además, se debe procurar que ningún grupo esté formado por dos letras iguales. Si esto sucediera, se puede adoptar la regla de insertar un carácter ficticio entre ellas, por ejemplo, de nuevo una *x*.

Para transformar un grupo se observa si las dos letras que lo forman están en la misma fila, en la misma columna o en diferente fila y columna del cuadro. Si ambas están en la misma fila o columna, se cifra cada grupo por el formado por las dos letras que siguen a cada una de ellas, respectivamente, en la misma fila o columna. Se considera que la letra que sigue a una que esté en la frontera final del cuadro es la que está al inicio del mismo en la misma fila o columna.

Si las dos letras se encuentran en filas y columnas diferentes, se dibuja mentalmente el rectángulo, que tiene como vértices opuestos las dos letras a cifrar y se considera que el

grupo cifrante que resulta es el formado por las dos letras que ocupan los otros dos vértices de dicho rectángulo. Se considera como primera letra del grupo cifrante la que está en la misma fila que la primera del grupo original. Para descifrar un mensaje bastará con aplicar las reglas inversas a las anteriores. Si se desea cifrar el mensaje "Wheatstone fue uno de los pioneros del telégrafo eléctrico", este se separa en grupos de dos letras: "wh ea ts to ne fu eu no de lo sp io ne ro sd el te le gr af oe le ct ri co" y se obtiene como criptograma el siguiente: "HS LR WV VP YC AP LY OP RC XU TN OV YC JY QN LC WZ CL KF DR YX CL ZS AF XN".

Rejillas de Julio Verne

El escritor Julio Verne (1828-1905) también hizo uso de criptosistemas en sus novelas. Así, en *El viaje al centro de la Tierra* (Verne, 2004) presenta un mensaje en un pergamino con caracteres rúnicos, correspondientes a un sistema de sustitución, que los protagonistas deben recuperar para seguir adelante con su aventura.

El sistema empleado en la novela *Matías Sandorf* (Verne, 1987) es más original, por lo que nos detendremos en él. Este método corresponde a un sistema de transposición que hace uso de unas rejillas.

Para transformar un mensaje, este se escribe en bloques formando tablas cuadradas de, por ejemplo, seis filas por seis columnas. Si el último bloque del mensaje no lo rellena, se añaden letras que lo completen. A continuación se diseña una rejilla con el mismo número de filas y columnas de modo que se agujerean nueve celdas de la rejilla (en general, si los bloques del mensaje son de tamaño $n \times n$, el número de agujeros de la rejilla es la cuarta parte de este tamaño).

Las posiciones de los agujeros de la rejilla no son cualesquiera. Estos han de distribuirse a lo largo de las filas y columnas de modo que las posiciones que ocupan no coincidan

con las de los agujeros cuando la rejilla se gira 90, 180 y 270 grados. Este diseño no es fácil, pero tampoco es demasiado complicado. A modo de ejemplo, a continuación se presenta una rejilla negra de 6×6 que verifica esta propiedad:

Una vez diseñada la rejilla, esta se coloca sobre el primer bloque del mensaje escrito y se escriben las nueve letras que quedan al descubierto como primera parte del mensaje cifrado. A continuación se gira 90º a la derecha la rejilla y se vuelve a colocar sobre el bloque del mensaje que se está trasformando y se anotan las letras que se ven como segunda parte del mensaje. El proceso continúa girando la rejilla otros 90º y luego los últimos 90º. Una vez que se tiene el mensaje cifrado, este se puede reescribir en el mismo formato que el mensaje original.

Si el mensaje a ocultar fuera "Julio Verne fue el autor de Matías Sandorf" este se escribiría de la siguiente manera:

j	u	l	i	o	v
e	r	n	e	f	u
e	e	l	a	u	t
o	r	d	e	m	a
t	i	a	s	s	a
n	d	o	r	f	x

de modo que al superponer la rejilla sobre el bloque anterior se obtendría el primer grupo de nueve letras: "J O R E E T D T S". Repitiendo este proceso otras tres veces,

después de girar la rejilla 90º cada vez, se obtendría el criptograma siguiente:

J O R E E T
D T S U I V
F L R M A R
N U A O A A
S D X L E E
U E I N O F

El receptor del mensaje deberá tener la misma rejilla que el emisor y repetir el mismo proceso que este para recuperar el mensaje original.

CAPÍTULO 2

Las máquinas cifradoras y la Segunda Guerra Mundial

Si bien los cifradores de rotación mediante discos de cifrado podrían considerarse como las primeras máquinas de cifrado, en este libro solo consideraremos como tales las que hacen uso de técnicas electromecánicas y no solo mecánicas o de desplazamiento (Mowry, 2014). Hoy en día se conoce la existencia de un gran número de máquinas cifradoras y hacer un repaso a todas ellas supone una tarea que va más allá del objetivo de este libro; es por ello que solo nos detendremos en algunas de las más significativas, como son la Hagelin, la Schluesselgeraet, la Sturgeon y, por supuesto, la Enigma, que requiere una sección aparte. Remitimos al lector interesado a las múltiples páginas de Internet donde se puede encontrar abundante información.

A partir de este capítulo, hablaremos de sistemas de cifrado, puesto que las transformaciones que sufren los mensajes para convertirse en criptogramas son complejas y podrían considerarse como operaciones, aunque no siempre sean matemáticas. Por esta razón, los mensajes se escribirán como textos normales (con tildes, espacios, etc.), salvo que el propio sistema de cifrado exija otro tipo de codificación previa al cifrado.

Las primeras máquinas

Las primeras máquinas cifradoras fueron desarrolladas por Boris Caesar Wilhelm Hagelin (1892-1983), todas ellas conocidas como Hagelin (Mowry, 2014), aunque hubo muchos modelos con diferentes nombres: C-35, C-36, C-37, M-209, BC-38, etc. En general, los números del modelo identificaban el año en que fueron construidas; así, la C-35 corresponde al modelo fabricado en 1935. Además, las máquinas con teclado propio se identificaban por una letra B antes del número de modelo.

El sistema de cifrado de las Hagelin se basaba en el cifrado del almirante británico Francis Beaufort (1774-1857), que a su vez era una modificación del cifrado de Vigenère (aunque en realidad el cifrado conocido como de Beaufort fue inventado en 1710 por Giovanni Sestri).

En el cifrado de Beaufort se utiliza la misma tabla que en el de Vigenère (tabla 6), pero en lugar de considerar como resultado de la cifra la celda en la que se cruzan las letras correspondientes a la letra a cifrar (columna) con la clave (fila), se considera como resultado la primera letra de la fila en la que se encuentra la letra de la clave que está en la columna de la letra a cifrar. Así, si con la clave *FUR* se desea cifrar la palabra "que", con el cifrado de Vigenère el resultado sería VOV, mientras que con el de Beaufort este sería PAN.

Dicho de otro modo: si en el sistema de Vigenère el texto cifrado fuera la "suma" de la clave con el texto claro (cifrado = clave+claro), en el de Beaufort sería la "resta" de la clave menos el texto claro (cifrado = clave–claro). La ventaja del método de Beaufort es que las operaciones de cifrado y descifrado son siempre una resta (claro = clave–cifrado), mientras que en el de Vigenère la operación de cifrado sería una suma, como ya se ha dicho, y el descifrado sería una resta (claro = clave–cifrado).

El modelo C-36 de la Hagelin tiene cinco rodillos, cada uno de los cuales tiene diferente número de pasadores (o

dientes): 17, 19, 21, 23 y 25. Cada pulsación de una letra hace que el rodillo gire y el pasador cambie, y dado que el número de pasadores de cada rodillo es primo con 26 (número de letras), resulta que su periodo de rotación es el máximo; esto es, hacen falta 3.900.225 (=17·19·21·23·25) cifrados de letras individuales para que se repita el ciclo de cifrado con los mismos rodillos.

Otro ejemplo de la Hagelin fue el modelo M-209, que era mucho más ligero y portátil. Se accionaba manualmente y fue utilizada por el ejército estadounidense durante la Segunda Guerra Mundial. La seguridad de la M-209 era relativamente buena para su época, a pesar de que ahora se conoce que, ya en 1943, los descifradores alemanes eran capaces de recuperar un mensaje en menos de cuatro horas. A pesar de ello, la M-209 siguió utilizándose por el ejército norteamericano durante muchos años más, pero siempre restringiendo su uso para mensajes tácticos (de hecho, fue usada hasta la guerra de Corea), para los que ese plazo de cuatro horas era más que suficiente.

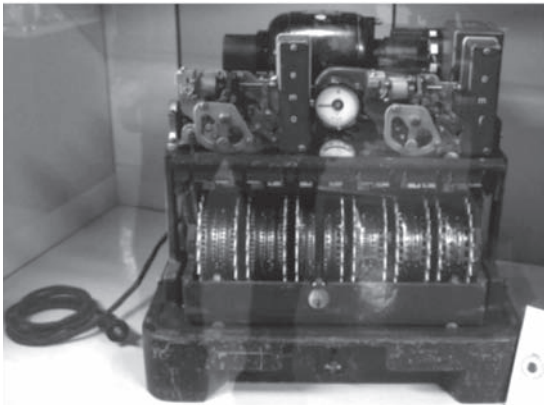
La Schluesselgeraet (dispositivo de cifrado) de la compañía Wanderer Werke es otra de las máquinas de cifrado que marcó un hito en el desarrollo de este tipo de dispositivos (Mowry, 2014), en especial el modelo SG-41 de 1941. La SG-41 fue desarrollada durante la Segunda Guerra Mundial por el inspector del Gobierno alemán Fritz Menzer como una posible sustituta de la Enigma. La SG-41 también fue conocida como el "Molino de Hitler" (*Hitlermühle*) debido a la gran manivela que tiene en su parte derecha.

La máquina se basa en el mismo principio que las Hagelin desarrolladas en Suecia, esto es, en rodillos con pasadores y no en el de su inmediata predecesora en el tiempo, la Enigma. En todo caso, supuso importantes mejoras con respecto a las Hagelin, lo que incrementó su seguridad y fueron muy difíciles de vulnerar. A ello contribuyó el hecho de que tuviera seis rodillos, que estos efectuaran movimientos muy irregulares y que pudieran realizar giros en ambos sentidos.

Máquinas para teletipos

La máquina Schlüsselzusatz, que significa "cifrado añadido", también fue conocida como Lorenz, siendo sus modelos más extendidos el SZ-40 y el SZ-42 (figura 8). Los criptógrafos británicos, que llamaban Fish (pescado) a todo el tráfico de los alemanes por teletipo, llamaron a esta máquina (y a su tráfico) Tunny (atún).

Figura 8
Modelo SZ-40 de la Tunny.



Fuente: Fotografía tomada por el autor en Bletchley Park.

La Tunny era una máquina electromecánica para el cifrado de teletipos basada en rodillos. Fue desarrollada por la compañía Lorenz y utilizada también durante la Segunda Guerra Mundial. Se empleaba en escenarios militares del más alto nivel entre Hitler y sus generales (Mowry, 2014).

El sistema de cifrado empleado en la Tunny era un método aditivo inventado en 1918 por el ingeniero americano Gilbert Sandford Vernam (1890-1960). Para entender este sistema, se debe tener en cuenta que las impresoras de teletipos no se basan ni en el alfabeto de 26 letras ni en el código Morse. Estas utilizan el código ideado por el ingeniero francés Jean Maurice Émile Baudot (1845-1903),

que consta de 32 símbolos y que también se conoce como Alfabeto Internacional de Telegrafía nº 1.

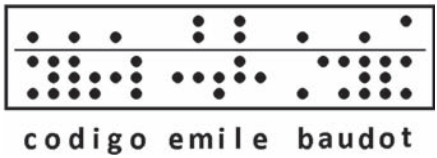
Cada una de las letras, números o símbolos se representa en el código Baudot mediante cinco marcas, que pueden ser agujeros o no, utilizando una máquina que tiene cinco teclas, dos de las cuales se presionan con la mano izquierda y tres con la derecha. Dado que las dos únicas opciones son presionar una tecla (hacer un agujero) o no, y como hay cinco posibles marcas, en total resultan $2^5 = 32$ símbolos, que son insuficientes para las letras del alfabeto, los números y los símbolos. Por ello se asignaron códigos especiales para distinguir si se utilizaba la tabla de letras o la de números y símbolos.

Aquí se muestran algunos ejemplos de codificación:

--•--: A	-• --•: B	-• •-•: C	-- •--: 1	-- -•-: 2	-- --•: 3

Actualmente, podríamos pensar en identificar los agujeros (•) con el 0 y un no agujero (-) con el 1, lo que equivaldría a utilizar 5 bits para cada letra, número o símbolo. Así, la letra A se codificaría, usando el código Baudot, como 11011, la letra B como 10110, etc. A modo de ejemplo, una cinta con la codificación de las palabras "codigo emile baudot" se muestra en la figura 9, teniendo en cuenta que la representación de cada letra se hace en vertical.

FIGURA 9
Ejemplo de código Baudot.



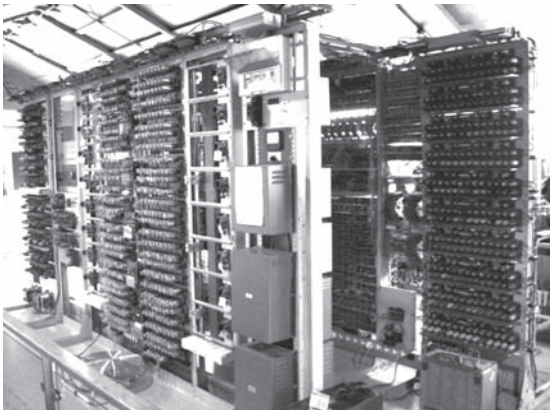
El sistema de cifrado de Vernam se explicará con detalle más adelante (véase "Cifradores en flujo" en el capítulo 4), pero por ahora basta decir que se trataba de enmascarar

cada una de las letras del mensaje original sumándole un carácter elegido de manera aleatoria, de modo que el resultado fuera la letra cifrada. Es evidente que el receptor del mensaje debería conocer esta secuencia de caracteres para restárselos al mensaje cifrado y recuperar el mensaje original.

Posteriormente, la máquina Tunny fue mejorada en dos ocasiones con nuevos modelos, el SZ-42a y SZ-42b, aunque su sistema de cifrado se rompió en Bletchley Park con la ayuda del Colossus, el primer ordenador electrónico-digital (figura 10).

Figura 10

Parte posterior de una reconstrucción del Colossus.



Fuente: Fotografía realizada por el autor en el museo de la computación de Bletchely Park.

Finalmente, la máquina conocida como Geheimschreiber fue desarrollada por Siemens & Halskeske en 1930, aunque su nombre oficial era SFM (*Schlüsselfernschreibmaschine* o teletipo de cifrado). Fue denominada Sturgeon (esturión) por los ingleses de Bletchley Park, probablemente siguiendo la misma pauta que la aplicada a la Tunny. De ella hubo varios modelos, aunque el más extendido fue el T-52 (Mowry, 2014). Este modelo también era una máquina de cifrado

electromecánico para las señales de teletipo y fue una de las máquinas de cifrado del ejército alemán más importantes durante la Segunda Guerra Mundial. Se utilizó a la vez que la Enigma y la Lorenz SZ-40. Finalizada la guerra, otros países seguirían utilizando la máquina, como Francia y los Países Bajos.

La Sturgeon era una máquina bastante pesada (alrededor de 100 kg) y su diseño básico consistía en una placa base con un teletipo Siemens T-36 en el centro. Detrás del teletipo estaba la unidad de cifrado que constaba de 10 ruedas dentadas. La máquina operaba directamente sobre las señales digitales de 5 bits del teletipo (a modo de código Baudot). Los mensajes en claro se imprimían de forma inmediata, pero los operadores de las máquinas no veían el texto cifrado a menos que utilizaran una clave errónea.

Inicialmente, los británicos tuvieron poco acceso a los mensajes cifrados con esta máquina, dado que cifraba teletipos; sin embargo, posteriormente algunos modelos fueron adaptados para las transmisiones de radio (Weierud, 2000).

La máquina Enigma

Sin duda, la máquina de cifrado por excelencia es la Enigma alemana (Singh, 2000: 131; Soler *et al.*, 2010). Hoy en día se pueden encontrar numerosos artículos, páginas web y programas informáticos donde no solo se explica su funcionamiento, sino que también se ofrecen simuladores de dicha máquina (en la bibliografía se incluyen algunos enlaces web donde encontrar ejemplos y simuladores de la Enigma). Aquí vamos a dar una visión genérica de la máquina y a presentar un ejemplo de cifrado y de descifrado con la misma.

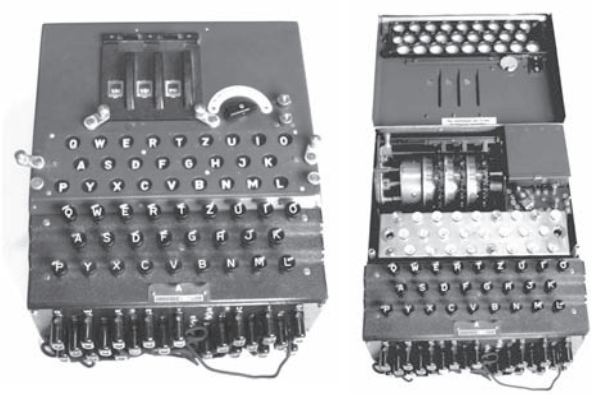
En 1918, el inventor alemán Arthur Scherbius (1878-1929) desarrolló una máquina, la Enigma (figura 11) para el cifrado de datos, especialmente para las comunicaciones bancarias y empresariales. Los primeros modelos de esta

máquina fueron patentados, pero no tuvieron mucho éxito hasta que el ejército alemán comenzó a utilizarlas a partir de 1925. La Enigma es, en esencia, una versión electromecánica del disco de Alberti.

Los tres elementos de que consta esta máquina (figura 11) son un *teclado* (similar al actual teclado QWERTY) para escribir el mensaje a cifrar, sobre el que se encuentra un *panel de lámparas* en el que se ilumina la letra que cifra cada una de las letras del texto original a medida que se van tecleando; sobre este está situado un *mecanismo de cifra,* que es el responsable del proceso de cifrado.

Figura 11

Máquina Enigma lista para ser utilizada (izquierda) y abierta (derecha).



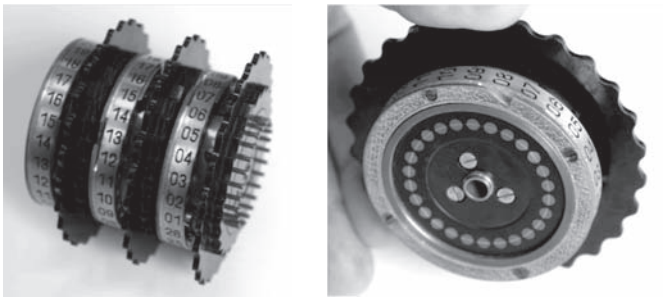
Fuente: Máquina Enigma cedida para ser fotografiada para este libro por el CCN. Fotografía realizada por J. Negrillo.

El mecanismo de cifra se basa en unos rotores (figura 12) conectados entre sí, cada uno de los cuales tiene 26 contactos que corresponden a las 26 letras del alfabeto. Cada uno de los rotores tiene 13 conexiones internas que enlazan dos a dos cada uno de sus 26 contactos y, además, posee unos conectores para hacer pasar la corriente de un rotor al siguiente. A continuación de los rotores, se coloca un reflector (que se puede elegir de entre dos posibles) que hace que la corriente

que pasa por los rotores se refleje en él y regrese de nuevo por los rotores para iluminar, finalmente, la lámpara que indica el resultado de la cifra.

FIGURA 12

Rotores de la máquina Enigma (izquierda) y detalle de un rotor (derecha).



FUENTE: MÁQUINA ENIGMA CEDIDA PARA SER FOTOGRAFIADA PARA ESTE LIBRO POR EL CCN. FOTOGRAFÍA REALIZADA POR J. NEGRILLO.

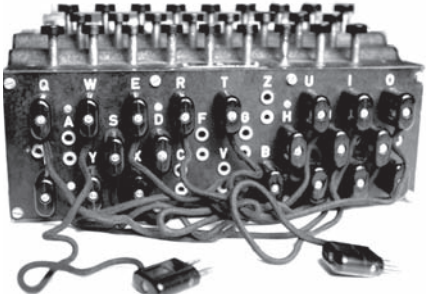
De este modo, cada vez que se pulsa una tecla para cifrar una letra, el sistema eléctrico de la máquina hace girar los rotores, produciendo la letra cifrada correspondiente. El resultado depende, entre otras cosas, de la posición que ocupa cada rotor con relación a los demás y de su propia colocación.

Además de las tres partes básicas ya mencionadas, las máquinas Enigma tienen un *clavijero* en la parte frontal de la máquina (figura 13), cuya función consiste en intercambiar, mediante cables, unas letras por otras; es decir, si se conectan las clavijas correspondientes a las letras A y H, por ejemplo, cada vez que se pulse la tecla A, la máquina cifrará como si se tratara de la letra H y viceversa (mantenemos el formato de las letras de la propia Enigma, cuyo teclado no distinguía mayúsculas de minúsculas).

De la Enigma se construyeron diferentes versiones, todas ellas con dos reflectores. Para la Wehrmacht la máquina usaba tres rotores (a elegir de entre cinco) y para la Kriegsmarine había dos modelos, uno que usaba tres rotores (a elegir de entre cinco normales más tres con doble muesca) y otro con

cuatro rotores (a elegir de entre cinco normales más tres con doble muesca y otros dos especiales, denominados β y γ).

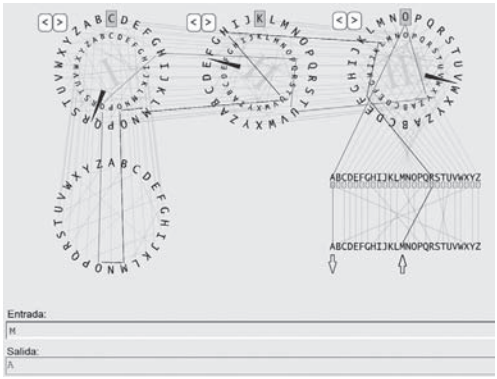
Figura 13
Clavijero.



Fuente: Máquina Enigma cedida para ser fotografiada para este libro por el CCN. Fotografía realizada por J. Negrillo.

A modo de ejemplo, siguiendo la figura 14, se puede describir el funcionamiento de la Enigma para cifrar una letra.

Figura 14
Simulación del cifrado de la letra M.



Fuente: Captura de pantalla del simulador incluido en el software Cryptool, en https://www.cryptool.org/

En primer lugar supondremos que los tres rotores que se han elegido son los denominados I, II y III y sus posiciones

son, respectivamente, izquierda, centro y derecha (I-II-III). Además, la colocación de cada uno muestra, en la posición 1, las letras C, K y N en su parte superior, respectivamente (más adelante se comentará la razón de que en la figura 14 la colocación del rotor III presente la letra O en lugar de la N, que está a su izquierda). Por último, en el clavijero se han conectado los siguientes pares de letras: BX, HP, MR, SU (nótese que también aparecen conectados sus pares simétricos XB, PH, RM y US).

En esta situación, si se desea cifrar la letra M, como esta está conectada mediante el clavijero con la letra R, la máquina supone que la letra que se va a cifrar es la R (camino oscuro de la figura 14). Ahora bien, como la letra R es la que ocupa la posición 18 en el alfabeto, el contacto del clavijero con el primer rotor de la derecha (el III) se produce en esta posición, por lo que la letra del contacto de entrada a este rotor es la que ocupa su posición 18, esto es, la F. Dado que la letra F tiene una conexión interna en el rotor III con la letra L, la letra de salida del rotor III hacia el segundo rotor (el II) es L, que ocupa la posición 24 del mismo. El contacto entre los rotores III y II se produce, por tanto, en la posición 24, que según la colocación que tiene el rotor II corresponde a su letra H. Ahora, la conexión interna del rotor II hace que la letra de entrada, H, produzca como salida la letra U, que ocupa la posición 11. Por su parte, la letra que ocupa la posición 11 del último rotor (el I) es la letra M, por lo que la entrada es precisamente esta letra. La letra de salida, por la conexión existente en este rotor, es la letra O.

Así, después de que la corriente atraviese los tres rotores, la letra M original se ha convertido en la letra O, que ocupa la posición 13 del rotor I. Como el rotor I está conectado con el reflector, las conexiones internas de este hacen que devuelva al rotor I la letra de la posición 15, es decir, la O (camino claro de la figura 14). Ahora se repite el proceso anterior entre los rotores, pero en sentido contrario, de modo que en el rotor I entra la letra de la posición 15, esto es, la Q, y sale la letra H (posición 6). En el rotor II entra la

letra P (posición 6) y sale la U (posición 11) hacia el rotor III, que tiene como letra de entrada laY (posición 11) y proporciona como salida la letra O (posición 1). En el clavijero, esta posición corresponde a la letra A, que al no estar conectada con ninguna otra es la letra resultante. En definitiva, el proceso anterior ha cifrado la letra M en la letra A.

Si este fuera el proceso completo, cada vez que se pulsara la letra M de un mensaje el resultado siempre sería la letra A, puesto que ni la posición ni la colocación de los rotores (ni del reflector ni del clavijero) cambiaría. No obstante, hay que tener en cuenta que cada vez que se pulsa una tecla, el primer rotor gira una posición hacia la izquierda (el clavijero y el reflector no cambian mientras se cifra un mensaje dado) y este giro es tal que, después de 26 pulsaciones, el primer rotor hace que el segundo rotor gire una vez y, después de 26 giros del segundo, el tercero gira una vez y así sucesivamente. Esta es la razón por la que en la figura 14 los rotores parecen mostrar originalmente las letras C, K y O, cuando en la configuración inicial dijimos que estas eran C, K y N. Lo que ha sucedido es que se muestra la posición de los rotores una vez que se ha pulsado la letra M, por lo que el rotor III ya ha girado una posición a la izquierda.

Una vez que se conoce la forma en la que la Enigma cifra cada una de las letras, veamos un ejemplo de cómo se cifraría un pequeño mensaje. Debe tenerse en cuenta que en el cifrado de la Enigma no se utilizan espacios para separar las palabras y el mensaje cifrado se presenta mediante grupos de cinco letras separados por espacios.

En primer lugar, se debe seleccionar la clave que se va a utilizar. Se debe tener en cuenta que los operadores de la Enigma disponían de un libro de claves que cambiaba, habitualmente, cada mes. En dicho libro se mostraban las claves a emplear para cada uno de los días del mes, ordenadas del último día al primero, de modo que se pudiera ir cortando (desde la parte inferior) las claves ya utilizadas en días anteriores.

A modo de ejemplo, una hoja conteniendo las claves de varios días de un mes podría ser la que se muestra en la tabla 7.

Tabla 7
Libro de claves de la Enigma.

DÍA	REFLECTOR	ROTORES	ANILLO	CLAVIJERO	IDENTIFICADOR
31	B	I-II-V	06 22 14	PO ML IU KJ NH YT GB VF RE DC	EXS TGY IKJ LOP
30	C	III-IV-II	17 04 26	BN VC XS WQ AZ GT YH JU IK PM	KIJ TFR BVC ZAE
29	B	V-I-III	15 02 09	ML KJ HG FD SQ TR EZ IU BV XC	QZE TRF IOU TGB
...	...	...	...	...	...

Si el día es el 30 del mes, por ejemplo, se debe observar la fila que señala tal día en la tabla 7. Así, dicha fila indica que el reflector que se debe colocar es el C; los rotores, ordenados de izquierda a derecha, son el III, el IV y el II. Además, las colocaciones de los rotores deben mostrar en su parte superior las letras que ocupan las posiciones 17, 4 y 26, esto es, las letras, Q, D y Z. La columna del clavijero señala los pares de letras que deben estar conectadas: BN, VC, etc. Finalmente, la columna del identificador no forma parte de la clave en sí, pero sirve para conocer el día en que se cifró el mensaje (y, por tanto, la clave que se utilizó). De hecho, cada identificador consta de cuatro grupos de tres letras cada uno y permite elaborar un indicador de la clave que se utilizará un día determinado. Este indicador está formado por cinco letras, tres de las cuales han de ser uno cualquiera de los grupos que aparecen en la celda del identificador correspondiente al día en que se va a cifrar un mensaje. A modo de ejemplo, para el día 30 serían válidos cualquiera de los siguientes indicadores: KIJOT, AKIJO, ATFRW, OABVC, AZAEL, etc. (se han subrayado los grupos de tres letras que corresponden a los identificadores del día 30).

Para cifrar un mensaje supongamos que, por ejemplo, el responsable de claves de ese día elige como indicador el grupo AKIJO. A continuación, debe elegir aleatoriamente dos grupos de tres letras, que podrán ser, por ejemplo, ABC y XYZ. El primer grupo se emplea como posición básica y el

segundo como clave para el mensaje a cifrar. La forma que tendrá el mensaje cifrado será la siguiente:

RECEPTOR DE EMISOR hhmm = 19 = ABC PQR =
AKIJO MENSA JECIF RADO =

En la primera línea aparecen los nombres del receptor y del emisor del mensaje separados por la palabra "DE" para indicar quién lo envía, seguidos de la hora y los minutos del momento en que se cifró el mensaje (hhmm), y entre dos separadores (=) el número total de letras de que consta el mensaje que se va a enviar (19). A continuación se incluyen el primer grupo de tres letras (ABC) que se eligió como posición básica y luego el resultado (PQR) de cifrar el segundo grupo o clave del mensaje (XYZ) habiendo colocado los rotores en la posición básica. La primera línea finaliza con un separador (=).

La segunda línea comienza con las cinco letras que representan el indicador que se eligió para ese día y mensaje (AKIJO), y a continuación el mensaje cifrado en grupos de cinco letras (a excepción del último si el número total no es múltiplo de 5). El mensaje cifrado finaliza con otro separador (=).

Supongamos que nuestro nombre es Alicia y deseamos cifrar el mensaje "la maquina alemana enigma" para ser enviado a Bernardo a las 18.30 horas del día 30 de un mes. El proceso que se debe seguir para ello es el siguiente[2]:

1. Elegir un identificador válido para el día 30: AZAEL.
2. Abrir la máquina, seleccionar y colocar adecuadamente el reflector, los rotores y sus posiciones, sus colocaciones (anillo), cerrar la máquina y conectar las letras en el clavijero.

2. Se recomienda utilizar un simulador de la Enigma para comprobar cada uno de los pasos siguientes, como, por ejemplo, el que se puede descargar desde la página http://users.telenet.be/d.rijmenants/en/enigmasim.htm

3. Generar al azar una posición básica de tres letras, sea TIC, y girar los rotores hasta esa posición (TIC = 20-09-03).
4. Generar al azar una clave de mensaje de tres letras, JLR, y teclearla para cifrarla, obteniendo: VDX.
5. Girar los rotores a la posición de la clave del mensaje (JLR = 10-12-18) y comenzar a teclear el mensaje en claro de 22 letras: la maquina alemana enigma.
6. La primera línea del criptograma será: BERNARDO DE ALICIA 1830 = 27 = TIC VDX =
7. La segunda línea es el texto cifrado (incluyendo como primera palabra el identificador del día): AZAEL QWFYX AJXVI PGNEA YURQW CV=

El proceso de descifrado que debe llevar a cabo Bernardo es el siguiente:

1. Observar las dos primeras líneas recibidas para saber quién envía el mensaje y si se es el destinatario, a qué hora se cifró, el número de letras que debe contener el mensaje cifrado (incluyendo las cinco del identificador), etc. Buscar en el libro de códigos el identificador, AZAEL, para conocer el día que se cifró el mensaje: día 30.
2. Abrir la máquina, seleccionar y colocar adecuadamente el reflector, los rotores y sus posiciones, sus colocaciones (anillo), cerrar la máquina y conectar las letras en el clavijero.
3. Girar los rotores hasta la posición básica mostrada en la primera línea del mensaje recibido, TIC (20-09-03).
4. Teclear la clave cifrada del mensaje que aparece en la primera línea del mensaje recibido, VDX, para descifrarla, obteniendo: JLR.
5. Girar los rotores a la posición obtenida, JLR (10-12-18) y empezar a teclear el texto cifrado para descifrarlo: lamaq uinaa leman aenig ma.

Hasta aquí un ejemplo de cómo cifra la máquina Enigma. Ahora vamos a analizar brevemente su seguridad, es decir, conocer por qué fue tan difícil romper su sistema de cifrado. Para ello, se debe conocer el número de posibles claves que podía utilizar. La Enigma más sencilla era la de la Wehrmacht, que tenía solo tres rotores, por lo que podemos empezar a contar:

1. Dado que se pueden seleccionar tres rotores de entre cinco posibles, hay combinaciones de 5 elementos tomados de 3 en 3: $5 \cdot 4 \cdot 3 = 60$ combinaciones.
2. Cada uno de los rotores puede ser cableado internamente de 26 maneras, lo que supone un total de $26^3 = 17.576$ conexiones diferentes.
3. Cada anillo se puede colocar en 26 posiciones, pero, como no hay rotores a la izquierda del tercero, solo los anillos del rotor derecho y del central afectan a los cálculos, lo que proporciona $26^2 = 676$ combinaciones más.
4. Si se utilizan 10 cables para el clavijero, resultan 150.738.274.937.250 conexiones posibles.

Todo ello da un total de $150.738.274.937.250 \cdot 60 \cdot 17.576 \cdot 676 = 107.458.687.327.250.619.360.000$ claves, es decir, la Enigma de la Wehrmacht tenía alrededor de $1{,}07 \cdot 10^{23} \approx 2^{77}$ claves, lo que es equivalente a utilizar claves de 77 bits.

Por su parte, llevando a cabo un cómputo similar para la Enigma de cuatro rotores empleada por la Kriegsmarine, resultan un total de 31.291.969.749.695.380.357.632.000 claves, es decir, la Enigma de la Kriegsmarine tenía alrededor de $3{,}1 \cdot 10^{25} \approx 2^{84}$ claves, lo que es equivalente a utilizar claves de 84 bits.

Descifrando la Enigma

Sin embargo, a pesar del número de claves tan enorme, fue posible romper el sistema empleado por la Enigma. Ello se debió, principalmente, a los investigadores polacos y,

posteriormente, a los británicos. Los principales criptógrafos polacos fueron el profesor de la Universidad Politécnica de Lwów (o Leópolis, hoy una ciudad ucraniana), Zdzislaw Krygowski (1872-1955) y tres exalumnos suyos: Marian Rejewski (1905-1980), Henryk Zygalski (1908-1978) y Jerzy Rózycki (1909-1942).

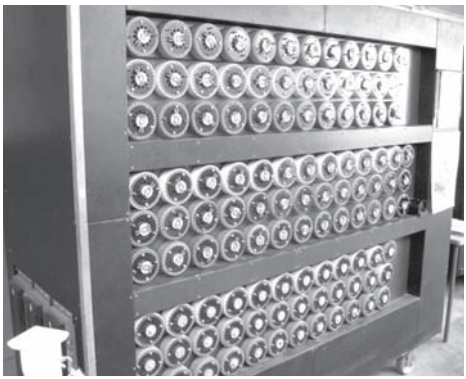
Rejewski utilizó los primeros modelos comerciales de la Enigma e hizo importantes descubrimientos matemáticos, como, por ejemplo, que ninguna letra se cifraba como ella misma, y determinó la permutación del tambor inicial. Posteriormente, diseñó las primeras máquinas llamadas Bombas (por el ruido que hacían) para probar automáticamente del orden de 26^3 posibilidades de las máquinas Enigma. Entre 1934 y 1938 se construyeron en Varsovia 17 máquinas que cifraban según la Enigma militar. De hecho, hasta 1939, el Gobierno polaco era capaz de descifrar las comunicaciones alemanas de las SS y de la Wehrmacht, pero luego las Enigmas alemanas fueron dotadas de más rotores, lo que hizo inútiles las réplicas polacas.

Los trabajos de los matemáticos e ingenieros polacos fueron retomados por los ingleses en Bletchley Park, en especial por el matemático Alan Turing (1912-1954), considerado hoy en día uno de los padres de las ciencias de la computación y el principal precursor de la Informática (De León *et al.*, 2014).

Alan Turing y Gordon Welchman (1906-1985) diseñaron una máquina electromecánica, conocida como la Bomba (llamada así en honor a la máquina diseñada por los polacos a pesar de que los principios en los que se basaba eran totalmente diferentes), que, ante un mensaje cifrado con la Enigma, permitía desechar una gran cantidad de posibles claves (figura 15). Este diseño estaba basado en los trabajos matemáticos de Turing relacionados con la llamada "máquina de Turing", donde demostró que, siempre que un problema matemático pudiera expresarse mediante un algoritmo, su máquina era capaz de implementar dicho problema (Turing, 1936). La forma de eliminar las claves inservibles se llevaba a

cabo mediante la implementación eléctrica de un sistema de deducciones lógicas que permitían detectar contradicciones en las claves y eliminarlas.

FIGURA 15
Reconstrucción de la Bomba de Turing.

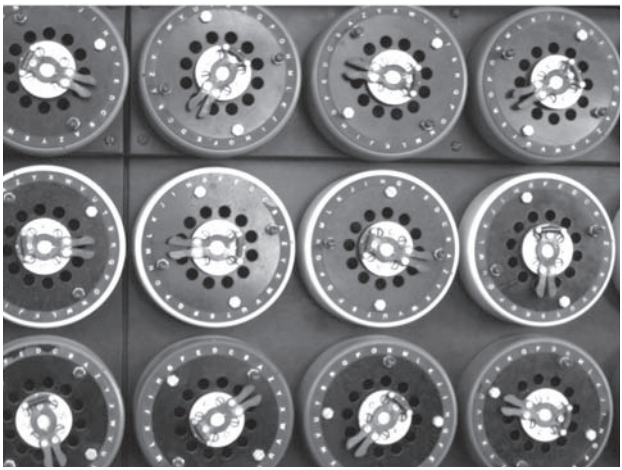


FUENTE: FOTOGRAFÍA REALIZADA POR EL AUTOR EN BLETCHLEY PARK.

De forma más precisa, el proceso para descifrar los mensajes de la Enigma consistía en determinar la configuración de la máquina para una pareja formada por una parte de un texto claro y su correspondiente texto cifrado, de modo que cuando se interceptaba un mensaje, se buscaban las denominadas *cribs* (cunas), que eran posibles fragmentos de texto claro sin cifrar dentro de un texto cifrado. Estos fragmentos podían ser expresiones del tipo *An der Oberbefehlshaber* (el comandante en jefe), *Eine Gruppe* (un grupo), *Es Lebe Den Führer* (viva el Führer) o cualquier otro trozo de texto. Una vez que se descubría un fragmento, las asociaciones entre las letras del texto cifrado y el texto claro se introducían en la Bomba. Entonces, la Bomba, que contenía un gran número de tambores (figura 16), cada uno de los cuales replicaba un posible rotor de la Enigma (recuérdese que cada rotor unía parejas de letras mediante contactos internos, desconocidos para los criptoanalistas), analizaba todas las posibles configuraciones de las claves

para tratar de localizar las que correspondían a los trozos de texto claro y texto cifrado introducidos. Una vez que se encontraban estas configuraciones, todos los mensajes cifrados con esta configuración podían ser descifrados.

FIGURA 16
Detalle de los tambores usados en la Bomba de Turing.



FUENTE: FOTOGRAFÍA REALIZADA POR EL AUTOR EN BLETCHLEY PARK.

En todo caso, ahora se sabe que la información recuperada por los criptoanalistas de Bletchley Park (a la que se asignó el código ULTRA) jugó un papel fundamental durante la Segunda Guerra Mundial y fue utilizada con extremo cuidado (de hecho, tal información nunca se empleó a no ser que fuera confirmada por una segunda fuente) con el fin de evitar las sospechas del mando alemán, que podría haber cambiado de sistema de cifrado, dejando de utilizar la Enigma y echando por tierra todo el trabajo de los criptoanalistas británicos.

En 2105 se estrenó la película *The Imitation Game* (traducida en España como *Descifrando Enigma*), en la que se presentan los trabajos de Turing en Bletchley Park para vulnerar los cifrados alemanes con la máquina Enigma.

Los Windtalkers

Los Windtalkers o Mensajeros del viento (Singh, 2000, cap. 5) fueron indios americanos de la tribu de los navajos, ubicados en los estados de Arizona, Nuevo México, Utah y Colorado, que participaron en la Segunda Guerra Mundial protegiendo las comunicaciones del ejército americano en el Pacífico.

Al parecer, con motivo de una visita de indios navajos al presidente Theodore Roosevelt, para tratar sobre sus condiciones de vida, este observó que el lenguaje de los indios era absolutamente incomprensible y pensó que podría utilizarse como medio de cifra, es decir, bastaría con considerar como texto cifrado el resultado de traducir al idioma de los indios navajos un texto en claro determinado. Bastaba con que cada batallón de las tropas americanas en el Pacífico empleara a un par de indios navajos en las transmisiones de radio para que las comunicaciones fueran seguras. De hecho, este sistema, conocido como "código navajo", jamás fue descifrado por los japoneses.

Esta idea no era la primera vez que se utilizaba. Así, durante la Primera Guerra Mundial, miembros de la tribu Choctaw, situados en la zona de los estados de Oklahoma y Alabama, fueron empleados como operadores de radio en el norte de Francia. El problema era que su idioma, llamado *muskogi*, no tenía palabras que fueran equiparables a los términos empleados en la guerra y el uso de las mismas podría dar lugar a malas interpretaciones por parte del receptor de radio.

Sin embargo, poco después del bombardeo de Pearl Harbor, varios indios navajos comenzaron un curso de formación para conocer la asignación de los términos modernos de la guerra con palabras del lenguaje navajo, que claramente no existían.

De este modo, se elaboró una especie de código por el que los indios navajos relacionaron palabras de su lengua con términos empleados en ambientes militares. Así, a modo de ejemplo, los comandantes eran jefes de guerra, los morteros

eran cañones que se agachan, un avión de caza era un colibrí o un avión de bombardeo en picado era un halcón comepollos. Este nomenclátor constaba de 274 palabras, que los navajos tuvieron que aprender de memoria para evitar que, si se escribían en un libro, este pudiera caer en manos enemigas.

Tabla 8

Equivalencia entre letras, palabras e idioma navajo.

LETRA	PALABRA	TRADUCCIÓN	NAVAJO
H	*Horse*	Caballo	Lin
I	*Ice*	Hielo	Tkin
T	*Turkey*	Pavo	Than-zie
L	*Lamb*	Cordero	Dibeh-yazzi
E	*Elk*	Alce	Dzeh
R	*Rabbit*	Conejo	Gah

No obstante, el uso de 274 palabras no era suficiente para resolver el problema, por ejemplo, de los nombres propios de personas, ciudades o ríos. Por ello, se elaboró otro código que asignaba a cada letra del alfabeto una palabra en inglés y esta se correspondía con otra palabra en navajo. Por ejemplo, Hitler sería cifrado en navajo de la siguiente manera: "Lin Tkin Than-zie Dibeh-yazzi Dzeh Gah" (tabla 8).

En 1968, el código navajo fue desclasificado y en 1982 el Gobierno de Estados Unidos les rindió homenaje declarando el 14 de agosto "Día nacional de los mensajeros de código navajo". Posteriormente, su participación en la Segunda Guerra Mundial fue adaptada al cine, en 2002, con la película *Windtalkers*.

CAPÍTULO 3

La criptografía de hoy

La criptografía utilizada a partir de la Segunda Guerra Mundial, sobre todo desde el uso extendido de los ordenadores personales, no tiene nada que ver con la que hemos visto hasta ahora. Su desarrollo y aplicaciones han crecido de tal forma que han llegado a instalarse en muchas de nuestras acciones cotidianas. En este capítulo veremos los inicios de este cambio.

Los nuevos paradigmas criptográficos

La diferencia fundamental entre la criptografía que hemos presentado hasta ahora y la que ha comenzado a desarrollarse y utilizarse desde mediados de los años setenta del siglo XX es que aquella tenía un carácter puramente militar y diplomático, es decir, se diseñaba para ser utilizada en estos ambientes, cuyas características son muy peculiares, mientras que la de ahora está presente en muchos de los entornos de la sociedad de la información. De hecho, hoy casi todos empleamos, de una u otra forma, algún dispositivo o sistema que utiliza la criptografía, aunque no seamos conscientes de ello. En el capítulo 6 veremos algunos de estos usos, pero baste decir que hay criptografía cuando empleamos la tarjeta de crédito,

cuando nos conectamos a nuestro banco a través de Internet, cuando hacemos compras electrónicamente, cuando firmamos electrónicamente algún documento con un certificado digital o con el Documento Nacional de Identidad electrónico (DNIe) y, en fin, cuando enviamos y recibimos correos electrónicos o navegamos por Internet.

Pero no solo ha cambiado el tipo de ambiente en el que se utiliza, también lo han hecho algunos de sus principios fundamentales. El más importante de ellos se conoce como "principio de Kerckhoffs".

Hasta ahora, una de las premisas más extendidas de los sistemas criptográficos era que el adversario o enemigo no debía tener acceso a la clave que pudiera utilizarse (ni al mensaje original, desde luego) ni tampoco debía conocer el sistema empleado para cifrar la información. Dicho de otro modo: si los ejércitos aliados de la Segunda Guerra Mundial hubieran tenido a su disposición las máquinas de cifrado de los alemanes, habrían vulnerado los sistemas con mayor facilidad. Pero es que si tal hecho hubiera sucedido y hubiera sido conocido por los alemanes, estos habrían cambiado el sistema de cifrado inmediatamente, haciendo inútiles los trabajos de criptoanálisis de los aliados.

A pesar de que el principio de Kerckhoffs fue publicado en 1883 por el lingüista y criptógrafo holandés Augusto Kerckhoffs von Nieuwenhof (1835-1903), en su libro *La Cryptographie militaire*, no ha llegado a convertirse en el paradigma fundamental de la criptografía moderna hasta hace solo unos años, cuando la técnica del cifrado y descifrado de la información ha pasado a convertirse en una ciencia.

El principio de Kerckhoffs establece lo siguiente: "La seguridad de un criptosistema no debe depender de mantener secreto el algoritmo de cifrado empleado. La seguridad depende solo de mantener secreta la clave". Dicho de otro modo: hoy en día no es, en absoluto, recomendable utilizar un sistema de cifrado que no haya sido analizado y validado por la comunidad científica, de modo que se esté seguro (hasta donde esto sea posible) que tal sistema es invulnerable.

Por tanto, siempre se debe dar por hecho que el adversario conocerá todas las características del criptosistema, es decir, cómo se generan las claves, qué longitud y propiedades tienen, cuál es el proceso de cifrado y de descifrado y qué algoritmos son los más eficientes para llevarlo a cabo. Además, se debe suponer que el enemigo es capaz de acceder al texto cifrado, que posee ordenadores tan potentes como los de uno y que es, al menos, tan inteligente como quien diseñó el sistema que estamos usando. Lo único que se supone que el adversario no conoce es ni el texto en claro ni la clave que se utilizó para cifrarlo.

Como conclusión se puede afirmar que si una organización utiliza sus propios sistemas de cifrado sin que hayan sido contrastados por personal experto y ajeno a la organización, entonces, con un alto grado de probabilidad, tal sistema será vulnerable.

Por lo señalado anteriormente, lo recomendable es utilizar los sistemas de cifrado que se han convertido en estándares porque de ellos se conocen todas sus características, incluyendo el tiempo y los recursos computacionales que hacen falta para romperlos. De este modo, uno puede estar seguro de que está empleando los sistemas más eficientes y más seguros. Estos se han convertido en estándares porque han sido analizados por la comunidad científica y han sido, posteriormente, propuestos para su uso como normas internacionales (Menezes *et al.*, 1997; Fúster *et al.*, 2012).

Criptografía simétrica y asimétrica

Como ya hemos mencionado, debido a la adopción del principio de Kerckhoffs, lo que hasta ahora hemos conocido como una técnica para el cifrado y descifrado de información se ha llegado a convertir en una ciencia en la que participan expertos en ciencias de la computación (ingenieros informáticos y de telecomunicación, principalmente), matemáticos y físicos.

Los sistemas de cifrado que hemos presentado hasta ahora tienen la particularidad de que tanto el emisor como el receptor utilizan la misma clave ya sea para cifrar o para descifrar, si bien el proceso de descifrado puede ser ligeramente diferente al de cifrado, pero no sustancialmente. Cuando sucede esto, se dice que el sistema de cifrado es de *clave simétrica* o de *clave secreta*. La razón es que en ambos extremos de la comunicación la clave es la misma y esta debe mantenerse en secreto por los dos usuarios, no debiendo compartirla con nadie. Si la clave se llegara a conocer, todos los mensajes cifrados con dicha clave podrían ser descifrados, quedando así anulada la confidencialidad de la comunicación.

Existen dos tipos de cifradores de clave simétrica, que serán analizados en el capítulo siguiente: los cifradores en flujo y los cifradores en bloque.

Esta criptografía simétrica es la que se ha venido empleando desde siempre (no había otra), hasta que a finales de los años setenta se publicó un artículo que mostraba cómo era posible que dos personas se pudieran intercambiar información en presencia de un adversario sin que este pudiera llegar a conocer qué información era la intercambiada. El artículo de Whitfield Diffie (1944-) y Martin Hellman (1945-) dio lugar a lo que más tarde se conocería como criptografía de *clave asimétrica* o de *clave pública* (Diffie y Hellman, 1976). En esta criptografía, cada usuario posee dos claves: una pública y otra privada. La primera la da a conocer a todos aquellos de los que desea recibir mensajes cifrados y es la que estos utilizan para cifrar la información que le van a enviar. La segunda la guarda en secreto y es la que le permite descifrar la información recibida. Como se puede deducir, ambas claves deben estar relacionadas para que cada una de ellas invierta el proceso realizado por la otra, pero esta relación ha de ser lo suficientemente compleja como para que sea imposible o, al menos, muy difícil, deducir la clave privada a partir del conocimiento de la pública.

Esta complejidad que relaciona ambas claves suele estar basada en un problema matemático difícil de resolver. Por

este motivo, los criptosistemas asimétricos se clasifican en función del tipo de problema matemático en el que se basa su seguridad: de la factorización entera, del logaritmo discreto, etc. El problema de la factorización entera consiste en determinar los factores o divisores de un número dado, si los tiene, mientras que el problema del logaritmo discreto consiste en calcular logaritmos en conjuntos finitos. Los principales problemas matemáticos empleados en la criptografía asimétrica se comentarán con más detalle a medida que vayan apareciendo en las descripciones de los criptosistemas. En todo caso, hay una serie de términos y conceptos matemáticos, en su mayoría elementales, cuyo conocimiento es indispensable para entender la criptografía moderna.

Algunos conceptos matemáticos

En esta sección presentaremos, de forma resumida, los conceptos matemáticos más importantes para entender la criptografía moderna (Menezes *et al.*, 1997; Fúster *et al.*, 2004; Durán *et al.*, 2005).

En primer lugar, se debe aclarar el concepto de *dificultad* a la hora de resolver un problema. En general, consideramos que un problema es difícil, sobre todo si es de matemáticas, cuando no sabemos cómo resolverlo. Sin embargo, en criptografía se considera que un problema es difícil si se tarda mucho tiempo en obtener la solución, aunque se sepa cómo se resuelve y se disponga de un algoritmo que la determine. De hecho, la mayor parte de los problemas matemáticos que se emplean en criptografía se sabe cómo se resuelven y, además, se dispone de algoritmos y programas que los implementan y son capaces de resolverlos. Sin embargo, obtener la solución para un caso real de uno de estos problemas suele requerir más de 2^{80} años, es decir, más de la edad del universo. De ahí que se diga que la dificultad del problema es de tipo *computacional.*

Por otra parte, las operaciones matemáticas que se realizan en criptografía suelen hacer uso exclusivo de números

enteros, representados por la letra **Z**, no del resto de números reales. En otras palabras, no se emplean números con cifras decimales. Esto es así porque si utilizamos números con decimales para cifrar un texto, como el proceso de cifrado requiere hacer determinadas operaciones con tales números, y como la aritmética que emplean los ordenadores tiene una precisión finita, el descifrado no proporcionará el mensaje original por la acumulación de errores en las operaciones con decimales.

Así pues, y de forma resumida, se trata de elegir conjuntos de números enteros en los que sea posible realizar las cuatro operaciones básicas: suma, resta, multiplicación y división, de modo que se verifiquen algunas propiedades elementales.

Si se considera un conjunto de números enteros con una operación (la suma o la multiplicación, para concretar), se desea que, al operar dos números, el resultado siga perteneciendo al mismo conjunto (la operación es interna), de modo que al operar tres elementos ordenados el resultado sea el mismo, no importando si se operan los dos primeros y luego el tercero o si se opera el primero con el resultado de operar los dos últimos (la operación es asociativa). Además, debe existir un elemento distinguido (llamado neutro) tal que cualquier otro número operado con él proporcione como resultado el mismo número (en la suma este elemento es el 0 y en la multiplicación es el 1). También es necesario poder realizar la operación inversa a la considerada (resta con la suma y división con la multiplicación); para ello es obligatorio que, dado un elemento cualquiera del conjunto, exista otro (llamado opuesto en la suma e inverso en la multiplicación) que, al ser operado con el primero, el resultado sea el neutro.

Un conjunto en el que se considera una operación con todas estas propiedades se abrevia diciendo que es un grupo aditivo (si la operación es la suma) o multiplicativo (si es la multiplicación).

Si la operación es, además, conmutativa, es decir, si el resultado no depende del orden de los elementos, el grupo se

llama conmutativo. Un grupo conmutativo con un número infinito de elementos es el conjunto de los números enteros con la suma. Sin embargo, este conjunto con la multiplicación no es un grupo porque ningún número entero, salvo el 1, posee inverso dentro del mismo conjunto.

Retomando la cuestión de los conjuntos de números donde se puedan definir y llevar a cabo operaciones cuya aritmética no produzca errores debido al redondeo, lo cierto es que la respuesta la tenemos más cerca de lo que pudiera parecer. Aunque parezca sorprendente, esta aritmética elemental la realizamos de forma habitual y no le prestamos, en general, mucha atención porque la tenemos completamente asumida e interiorizada.

En primer lugar, pensemos en las operaciones y números que llevamos a cabo con las horas y los minutos: lo que se llama la "aritmética del reloj".

Nadie se sorprende si en lugar de oír "quedamos a las 17" oye "quedamos a las 5". Todos entendemos que es la misma hora, esto es, que $17 = 5$, y lo que es más, nunca usamos horas mayores de las 12, que, dicho sea de paso, identificamos con las 0 horas al terminar el día ($12 = 0$).

Algo parecido podemos decir de los minutos y los segundos: nunca pasamos de 60; de hecho, para contar minutos solo consideramos números enteros entre el 0 y el 59. El minuto siguiente al 59 es el 0, y dos minutos después del 59 es el minuto 1, y así continuamos. En definitiva, estamos diciendo que $59+1 = 0$ o que $59+2 = 1$, o lo que es igual, identificamos el minuto 60 con el 0.

Abundando en estas cuentas tan "raras" que hacemos, podemos pensar en que si son las 5 y quedamos en vernos con una persona dentro de 74 horas, lo que hacemos es ir restando a 74 los días que sean precisos para saber a qué hora hemos quedado. Esto es, vamos restando a 74 el número 24 (las horas que tiene un día) tantas veces como podamos para saber cuántas quedan y sumarlas a 5 para saber la hora de la cita. Dado que $74 = 24+24+24+2 = 24\cdot3+2$, resulta que hemos quedado a las 7 (dos horas más tarde de las 5), dentro de 3 días.

En definitiva, lo que hemos hecho es considerar que $5+74 = 5+2 = 7$, pero dentro de tres días. Ahora estamos considerando (mentalmente) que $24 = 0$ porque al terminar un día volvemos a comenzar desde la hora 0.

Algo similar sucede cuando consideramos los grados de una circunferencia. Sabemos que la circunferencia tiene 360^o, por lo que después de una vuelta volvemos al lugar de partida. De hecho, somos conscientes de que dar un giro de 360^o es quedarse en la misma posición de la que se partió y que si giramos 400^o es como si hubiéramos girado solo 40^o, dado que de nuevo identificamos 360 con 0. De hecho, para saber cuántos grados equivalen a 12.345^o basta con dividir tal cantidad entre 360 (una vuelta) para saber cuántas vueltas hemos dado a la circunferencia, de modo que lo que quede de esa división (el resto) será el ángulo que es equivalente a la cantidad inicial. Como $12.345 = 360 \cdot 34+105$, resulta que 12.346^o es equivalente a 105^o. Yendo un poco más lejos, si sumamos cuatro veces un ángulo recto, lo que obtenemos es: $90+90+90+90 = 4 \cdot 90 = 360 = 0$, esto es, obtenemos como resultado 0 al multiplicar dos números (4 y 90), ninguno de los cuales es 0.

Así pues, existen numerosas situaciones (reloj, días, circunferencia, etc.) en las que realizamos operaciones aritméticas un tanto extrañas, pero que no nos sorprenden.

Ahora bien, la cuestión que nos planteamos ahora es: ¿por qué nos limitamos a igualar a 0 los números 12, 24, 60 o 360? ¿Qué problema habría en considerar otros números e identificarlos con el 0? Las operaciones tendrían las mismas normas que con el reloj, pero habríamos conseguido una generalización, que desde el punto de vista matemático es tan posible y real como cualquier otra.

Las operaciones anteriores han consistido en dividir un número dado entre el valor que tomamos como 0 (12, 24, 60, etc.) y considerar como resultado el resto de esa división. Pues bien, esta forma de operar se conoce en matemáticas como "aritmética modular", siendo el *módulo* el número entre el que dividimos, por ejemplo n, y se escribe abreviadamente

como (mod n). El conjunto de los números que se utilizan cuando se hace módulo por uno de ellos, sea n, se representa por $\mathbf{Z}_n$ y sus elementos son los números comprendidos entre el 0 y el n–1, dado que son los únicos restos posibles cuando se divide entre n. En definitiva, a modo de ejemplo, se verifica que:

$8+7 = 15 \equiv 3 \pmod{12}$, $10+5 = 15 \equiv 2 \pmod{13}$,
$4+10 = 14 \equiv 0 \pmod{7}$

Debe hacerse notar que en lugar de escribir $15 = 3 \pmod{12}$, en general se utiliza el símbolo $\equiv$ para resaltar que lo que se escribe no es una igualdad estricta en el sentido tradicional, sino una igualdad módulo un número, de modo que se escribe $15 \equiv 3 \pmod{12}$.

Si en lugar de sumar, la operación que llevamos a cabo es el producto o la resta, la forma de proceder es análoga: se lleva a cabo la operación indicada y luego se hace módulo por el número correspondiente:

$8 \cdot 7 = 56 \equiv 8 \pmod{12}$, $10 \cdot 5 = 50 \equiv 11 \pmod{13}$,
$4 \cdot 10 = 40 \equiv 5 \pmod{7}$, $8-7 \equiv 1 \pmod{12}$, $10-5 \equiv 5 \pmod{13}$,
$4-10 = -6 \equiv 7-6 \equiv 1 \pmod{7}$

En definitiva, se puede afirmar que $\mathbf{Z}_n$ es un grupo aditivo conmutativo, cualquiera que sea n.

La división es algo más compleja porque para poder dividir un número entre otro, debe existir el inverso del segundo, es decir, calcular a/b es lo mismo que calcular $a \cdot b^{-1}$, siendo $b^{-1} = 1/b$ el inverso de b, es decir, el número que multiplicado por b da como resultado 1: $b \cdot b^{-1} = 1$. En los conjuntos con los que estamos trabajando solo existe el inverso de un número si este es primo con el módulo, es decir, si ambos números no tienen divisores en común (su máximo común divisor, denotado por mcd, es 1).

Dado que todos los restos que se obtienen cuando se divide un número entero cualquiera entre un número primo,

p, son menores que p, resulta que todos ellos son primos con p y así, todos, salvo el 0, tienen inverso. Dicho de otro modo, en $\mathbf{Z}_p$ es posible dividir entre cualquier número (salvo el 0). Sin embargo, esto no sucede cuando se considera un número compuesto, n, dado que al tener divisores, en $\mathbf{Z}_n$ no siempre existe el inverso de un número dado. A modo de resumen, se puede decir que si en el conjunto $\mathbf{Z}_p$ se consideran las operaciones de suma y multiplicación, resulta que se cumplen las siguientes propiedades: la multiplicación es distributiva respecto a la suma, es un grupo aditivo conmutativo y, si no se considera el 0, es un grupo multiplicativo conmutativo. Todo ello se resume diciendo que $\mathbf{Z}_p$ es un *cuerpo* conmutativo. Esto no se puede afirmar de $\mathbf{Z}_n$, dado que, en este caso, no se verifica la tercera condición.

No obstante, es interesante analizar los elementos de $\mathbf{Z}_n$ que sí tienen inverso Así, el subconjunto de $\mathbf{Z}_n$ de los números que tienen inverso se denota por $\mathbf{Z}_n^*$. A modo de ejemplo, se tiene que

$$\mathbf{Z}_7^* = \{1,2,3,4,5,6\}, \mathbf{Z}_{12}^* = \{1,5,7,11\}$$

Dado que en $\mathbf{Z}_7^*$ se verifica que $3 \cdot 5 = 15 \equiv 1 \pmod 7$, resulta que 3 y 5 son inversos entre sí, esto es, $3^{-1} \equiv 5 \pmod 7$ y $5^{-1} \equiv 3 \pmod 7$. Por otra parte, $7 \cdot 7 = 49 \equiv 1 \pmod{12}$, es decir, 7 es su propio inverso. Así pues, si queremos calcular, por ejemplo, 4/3 (mod 7) y 5/7 (mod 12), bastará con calcular $4/3 = 4 \cdot 3^{-1} \equiv 4 \cdot 5 \equiv 20 \equiv 6 \pmod 7$, $5/7 = 5 \cdot 7^{-1} \equiv 5 \cdot 7 \equiv 35 \equiv 11 \pmod{12}$.

El número de elementos que tiene el conjunto $\mathbf{Z}_n^*$ es el número de números primos con n y menores que n. Este valor se representa por $\phi(n)$ y se conoce como "indicador de Euler". Por tanto, el número de elementos del conjunto $\mathbf{Z}_p^*$ es $\phi(p) = p-1$, cuando p es primo; mientras que si n no es primo, el número de elementos de $\mathbf{Z}_n^*$ es $\phi(n)$, que es mucho más difícil de calcular, dado que hay que determinar todos los números que son primos con n.

En el apartado "Transformación de mensajes: criptografía" de la Introducción se presentó un ejemplo de cifrado que consistía en multiplicar un número por 23, dividir el producto obtenido entre 256 y considerar como resultado el resto de la división. El descifrado consistía en multiplicar el cifrado por 167 y considerar como resultado el resto de la división del producto obtenido entre 256. Como ejemplo se cifraba el número 99. Con lo dicho anteriormente, ahora queda claro que la operación que se hacía para cifrar era multiplicar por 23 (mod 256), mientras que se descifraba haciendo 167 (mod 256). Dado que $23 \cdot 167 = 3841 \equiv 1 \pmod{256}$, es decir, 23 y 167 son inversos módulo 256, resulta que las operaciones de cifrado y descifrado son, en este caso, una la inversa de la otra.

Como ya hemos dicho, los conjuntos de la forma $\mathbf{Z}_p$ son muy importantes en criptografía porque en ellos se pueden realizar tanto la suma como el producto, así como sus operaciones inversas respectivas: la resta y la división (salvo por 0).

Por otra parte, los grupos multiplicativos $\mathbf{Z}_p^*$ tienen otra propiedad que los convierte en lo que se denominan *grupos cíclicos*. Esta propiedad consiste en que algunos de los elementos de $\mathbf{Z}_p^*$ generan el resto de los elementos, es decir, existen elementos, llamados *generadores*, que al ser multiplicados por sí mismos de forma sucesiva, esto es, al ser elevados a las diferentes potencias, dan lugar a todos los demás números del conjunto. A modo de ejemplo, se puede ver que el 3 es un generador de $\mathbf{Z}_7^*$, dado que sus potencias sucesivas dan lugar a todos los elementos:

$3^1 \equiv 3 \pmod{7}$, $3^2 = 9 \equiv 2 \pmod{7}$, $3^3 = 27 \equiv 6 \pmod{7}$,
$3^4 = 81 \equiv 4 \pmod{7}$, $3^5 = 243 \equiv 5 \pmod{7}$,
$3^6 = 729 \equiv 1 \pmod{7}$

Como se puede ver, la última potencia anterior (que es el número de elementos del conjunto considerado) da como resultado 1, por lo que las potencias siguientes vuelven a repetirse, es decir, entran en un ciclo. Algo similar se podría

hacer con el 5, que es otro generador. Sin embargo, 2 no es un generador, dado que sus potencias sucesivas no dan lugar a todos los elementos del conjunto:

$2^1 \equiv 2 \pmod 7$, $2^2 \equiv 4 \pmod 7$, $2^3 = 8 \equiv 1 \pmod 7$,
$2^4 \equiv 2 \pmod 7$

Una de las operaciones más extendidas en criptografía tiene por objetivo calcular la potencia de un número entero elevado a otro, módulo un tercero, es decir, se trata de determinar el valor de la siguiente exponenciación modular:

$x = a^k \pmod n$

donde la base, a, el exponente, k, y el módulo, n, son, en general, grandes. El algoritmo más utilizado se conoce como *elevar al cuadrado y multiplicar*. A continuación vamos a presentar este algoritmo e ilustrarlo con un ejemplo, dado que será utilizado en más de una ocasión posteriormente.

La forma de proceder a partir del conocimiento de los números enteros, a, k y n, es la siguiente: se escribe el exponente en base 2 (binaria) y se considera un valor intermedio igual a la base: $x = a$. A continuación se repite un bucle que recorre todos los bits de la clave, desde la izquierda, comenzando por el segundo. En este bucle se calcula el cuadrado de x módulo n y si, además, el bit de la clave es 1, se multiplica x por a módulo n, en caso contrario, se continua con el siguiente bit de la clave hasta agotar el bucle. Este proceso, escrito en pseudocódigo, es como sigue:

1. Se determina la expresión en base 2 (binaria) del exponente (en este caso se supone que tiene r+1 bits): $k = (k_r\, k_{r-1}\, k_{r-2} \ldots k_1\, k_0)$.
2. Se hace la asignación: $x \leftarrow a$.
3. Se ejecuta el siguiente bucle:
   desde i igual a r–1 hasta 0 hacer
   a) Se hace la asignación: $x \leftarrow x^2 \pmod n$

b) Se analiza el siguiente condicional:
si ($k_i = 1$) entonces $x \leftarrow (x \cdot a) \pmod{n}$

4. Se devuelve el valor de x.

Este algoritmo es un algoritmo eficiente, es decir, necesita muy poco tiempo de computación.

A modo de ejemplo, veamos cómo se calcula el valor de $x \equiv a^{23} \pmod{n}$. Como $k = 23$ y dado que $23 = 2^4+2^2+2+1$, resulta que, al ejecutar el algoritmo anterior, se tiene lo siguiente:

1. Se calcula la expresión binaria de k: $k = (k_4\ k_3\ k_2\ k_1\ k_0) = (1\ 0\ 1\ 1\ 1)$.
2. Se asigna a x el valor de a: $x = a$
3. Desde i igual a 3 hasta 0 hacer
   a. $i = 3$
      i. $x \equiv a^2 \pmod{n}$
      ii. ¿Es $k_3 = 1$? No.
   b. $i = 2$
      i. $x \equiv (a^2)^2 \pmod{n}$
      ii. ¿Es $k_2 = 1$? Sí. Se calcula $x \equiv (a^2)^2 a \pmod{n}$
   c. $i = 1$
      i. $x \equiv ((a^2)^2 a)^2 \pmod{n}$
      ii. ¿Es $k_1 = 1$? Sí. Se calcula $x \equiv ((a^2)^2 a)^2 a \pmod{n}$
   d. $i = 0$
      i. $x \equiv (((a^2)^2 a)^2 a)^2 \pmod{n}$
      ii. ¿Es $k_0 = 1$? Sí. Se calcula $x \equiv (((a^2)^2 a)^2 a)^2)a \pmod{n}$
4. Se devuelve el valor de x: $x \equiv (((a^2)^2 a)^2 a)^2)a \pmod{n}$

Por tanto, se tiene que:

$$x = (((a^2)^2 a)^2 a)^2 a = ((a^4 a)^2 a)^2 a = ((a^5)^2 a)^2 a = (a^{10} a)^2 a$$
$$= (a^{11})^2 a = a^{22} a \equiv a^{23} \pmod{n}$$

Hasta aquí se han presentado los conceptos matemáticos que van a ser empleados en los capítulos siguientes y que son la base de la criptografía moderna.

CAPÍTULO 4

Criptografía de clave simétrica

Antes de presentar los diferentes sistemas de cifrado, hemos de señalar que, dado que en la actualidad las comunicaciones y la mayor parte de las operaciones se hacen a través de ordenadores, lo normal es modificar los mensajes y codificarlos mediante determinado número de bits, esto es, de ceros y unos. No importa si el mensaje es un texto, una imagen, un fichero de voz o de vídeo, toda la información se codifica en binario.

A modo de ejemplo, baste recordar el código de Baudot (apartado "Máquinas para teletipos" del capítulo 2), que ha sido el precursor del ASCII. Existen otros sistemas de codificación, cuyo uso depende, fundamentalmente, de cómo se va a tratar la información. Ejemplos de otros códigos son: Unicode y UTF-8 (8-bits *Unicode Transformation Format*).

Así pues, mientras no se diga lo contrario, se supondrá que el mensaje se ha codificado con bits utilizando un sistema de codificación conocido y aceptado.

Cifradores en flujo

En este tipo de cifrador se consideran los bits que representan al mensaje en claro codificado y se transforma cada uno de ellos dando lugar a otra colección de bits que forman el texto

cifrado. Tras la recepción del mensaje, se descifran uno a uno los sucesivos bits del criptograma recuperando el mensaje original codificado. El cifrado en flujo es el criptosistema más rápido y fácil de implementar de entre todos los cifradores, por lo que es utilizado en numerosas aplicaciones (Fúster *et al.*, 2012, cap. 2).

El "cifrado de Vernam" se puede considerar como el precursor de los cifradores en flujo. Como ya indicamos en el capítulo 2, se trata de enmascarar cada uno de los bits del mensaje original sumándole otro bit elegido aleatoriamente para obtener el mensaje cifrado.

La clave del cifrado de Vernam es una secuencia binaria perfectamente aleatoria de una longitud tan larga como el texto a cifrar y, además, se utiliza solamente una vez, de ahí que este procedimiento se denominase cifrado con cinta de uso único u OTP (*One Time Pad*).

La operación de cifrado es muy sencilla y se conoce como operación XOR, disyunción exclusiva o suma módulo 2. Esta operación solo considera bits y tiene la siguiente tabla de sumar: 0+0 = 0, 0+1 = 1, 1+0 = 1, 1+1 = 0, que puede resumirse mediante la siguiente regla: el resultado de la suma de dos bits iguales es 0; mientras que el de la suma de dos bits distintos es 1. Con esta definición, el cifrado de la palabra "Vernam" sería "s1XtÓ.", como se puede observar a continuación:

```
Texto claro:   01010110011001010111001001101110011000010110110 1
Clave:         00100101010101000010101000011010101100101101010 1
Texto cifrado: 01110011001100010101100001110100110100111011100 0
```

El descifrado se haría de la misma forma, esto es, se sumaría a la codificación del texto cifrado la clave para obtener el texto claro codificado en ASCII.

Debe tenerse en cuenta que la elección de la clave ha de hacerse de tal modo que el destinatario del texto cifrado debe poder repetirla exactamente para que recupere el mensaje original. Ahora bien, si esta clave, como ya hemos dicho, es completamente aleatoria, tan larga como la longitud del

mensaje y solo puede utilizarse una vez para que sea el sistema de cifrado absolutamente seguro, es claro que el sistema es muy poco eficiente y muy difícil de llevar a la práctica. Por esta razón, el cifrado de Vernam es solo una propuesta más bien teórica que apenas se emplea (se reserva para ambientes donde hace falta la máxima seguridad y con poca cantidad de información), pero que ha sentado las bases para los cifradores en flujo. El cifrado de Vernam es el único incondicionalmente seguro, esto es, el único cuya seguridad se ha demostrado matemáticamente. Eso no significa que el resto de los cifradores que se van a presentar posteriormente sean inseguros. De hecho, los cifradores modernos se consideran seguros en tanto no se pruebe que puedan ser vulnerados o que el problema matemático en el que se basa su seguridad pueda ser resuelto fácilmente.

Los cifradores en flujo son la alternativa práctica al cifrado de Vernam, pues son fácilmente implementables y con una seguridad elevada. En estos cifradores, el emisor y el receptor utilizan para cifrar una secuencia de bits que es pseudoaleatoria, esto es, que tiene las mismas propiedades que cabría esperar de una secuencia puramente aleatoria, pero que, por el contrario, puede ser reproducida de modo exacto cada vez que sea preciso. Esta secuencia se genera mediante un algoritmo público determinista, es decir, un algoritmo que siempre que se le proporciona la misma entrada produce la misma salida, y que es conocido por ambas partes. El algoritmo utiliza como entrada una *semilla*, que no es más que una clave corta compartida por emisor y receptor, habitualmente de 128-256 bits. Mediante este tipo de algoritmos, ambas partes pueden generar la misma clave de cifrado y descifrado compartiendo solo la semilla. La secuencia generada por dicho algoritmo se denomina "secuencia cifrante" y hace las veces de clave en el cifrado de Vernam. Por lo demás, los procesos de cifrado y descifrado son iguales que en el de Vernam.

Generadores de números pseudoaleatorios

Como acabamos de ver, la seguridad de los sistemas de cifrado en flujo depende de cómo sea de aleatoria o previsible la secuencia cifrante. Por ello, es fundamental que el diseño del algoritmo generador de esta secuencia se haga con todas las precauciones para garantizar que no sea previsible (Fúster *et al.*, 2012, cap. 2).

Decidir si una secuencia cifrante es segura para su uso criptográfico no es sencillo, dado que no existe un criterio único y universal que lo garantice. Sí que existen, no obstante, determinadas propiedades que toda secuencia cifrante tiene que cumplir para que pueda ser empleada en un cifrado en flujo.

En primer lugar, la secuencia debe tener un *periodo* tan grande como sea posible, es decir, el número de bits necesario para que la secuencia se repita debería ser, al menos, tan largo como la longitud de la secuencia a que da lugar el mensaje original. De hecho, esta propiedad no es difícil de cumplir en la práctica.

Por otra parte, la distribución de los ceros y los unos en la secuencia debe cumplir los llamados postulados de Golomb, publicados por Solomon Wolf Golomb (1932-) en 1982 (Golomb, 1982). Estos postulados vienen a exigir que los 0 y 1 de la secuencia aparezcan como sería esperable en una secuencia realmente aleatoria.

En tercer lugar, la secuencia cifrante debe ser *imprevisible*, es decir, si un adversario es capaz de obtener una parte de secuencia de determinada longitud, no debería poder predecir el bit siguiente a tal parte de la secuencia, con una probabilidad de acertar mayor que 0,5.

Existen diferentes tests que analizan si una determinada secuencia de bits puede ser utilizada con fines criptográficos. Entre ellas cabe destacar las del NIST SP 800-20 (el NIST es el National Institute of Standards and Technology norteamericano), las de Diehard y las de Tuftest, entre otros (Fúster *et al.*, 2012, cap. 2). En todo caso, se debe tener en cuenta que

las pruebas se diseñan para detectar las secuencias que no se deben utilizar porque no cumplen todas las propiedades que serían deseables. Dicho de otro modo, que una secuencia pase estas pruebas de pseudoaleatoriedad no garantiza que sea segura, es decir, pasar las pruebas es una condición necesaria de seguridad, pero no suficiente.

Finalmente, se pueden mencionar algunos cifradores en flujo que van más allá de una mera operación XOR, como pueden ser los conocidos como Rabbit, Salsa20/12, Sosemanuk, Grain o Trivium, por ejemplo (Fúster *et al.*, 2012, cap. 2).

Cifradores en bloque

Los cifradores en bloque son criptosistemas de clave simétrica que agrupan los bits del mensaje original en bloques de determinada longitud y luego cifran cada uno de tales bloques (Fúster *et al.*, 2012, cap. 3). Tienen la ventaja de que son involutivos, es decir, las mismas operaciones que se llevan a cabo para cifrar un bloque se repiten para descifrarlo, utilizando la misma clave en ambos procesos.

Para que un cifrador en bloque se pueda considerar viable, debe poseer determinadas propiedades. Destacamos las más importantes. Una de ellas se conoce como *dependencia entre bits*, y viene a significar que cada uno de los bits del texto cifrado debe ser una función compleja de todos los bits de la clave y de todos los bits del bloque de texto claro. Otra hace referencia al *cambio de los bits de entrada* en el sentido de que si se cambia un bit en un bloque del texto claro, aproximadamente la mitad de los bits de bloque cifrado correspondiente también deben cambiar. Algo similar debe suceder con el *cambio de los bits de clave*, es decir, el cambio en un bit de la clave producirá el cambio de la mitad de los bits del texto cifrado.

Por otra parte, todo cifrador en bloque suele estar formado por una serie de componentes como son: una *transformación inicial* de los bits de cada bloque, cuyo objetivo es, en

general, intentar hacer que el bloque sea aleatorio con el fin de disimular el aspecto de mensajes que pudieran estar muy formateados, como tablas, imágenes, etc. Esta transformación no suele tener significación criptográfica. La transformación inicial está acompañada por una *transformación final*, cuyo objetivo es invertir la primera y tiene lugar al final del proceso de cifrado de cada bloque.

El tercer componente es la *función de cifrado* propiamente dicha, que es iterada determinado número de veces. El número de iteraciones depende, en general, de la longitud del bloque.

La seguridad de un sistema recae, fundamentalmente, en su función de cifrado, por lo que su definición y justificación son básicas para garantizar la fortaleza del sistema. Algunos sistemas utilizan funciones de cifrado que hacen uso de determinadas transformaciones (en ocasiones denominadas cajas *S*), que no siempre están plenamente justificadas, lo que repercute en la fiabilidad del criptosistema.

Finalmente, todo cifrador en bloque debe tener un *algoritmo de expansión de clave*. Este algoritmo tiene por objeto generar un conjunto de subclaves, a partir de la clave original, que son las que van a ser utilizadas en las diferentes iteraciones de la función de cifrado. La clave original del usuario suele tener una longitud de entre 64 y 256 bits, dependiendo del cifrador en bloque que se utilice.

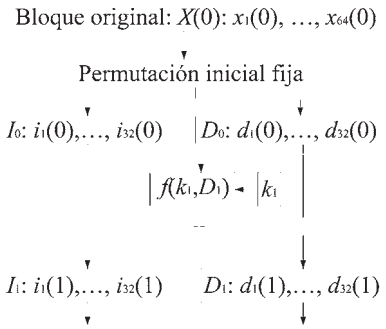
Los principales cifradores en bloque, cuyo uso actual está avalado por normas internacionales, son TripleDES y AES. Existen otros muchos cifradores cuya seguridad es similar a la de los anteriores, como son Twofish, Serpent, etc. (Fúster *et al.*, 2012, cap. 3).

El primer cifrador en bloque normalizado (por el NIST) para uso civil data del año 1977 y se conoce como DES (*Data Encryption Standard*) o DEA (*Data Encryption Algorithm*). Hoy en día este sistema ha quedado en desuso, puesto que se considera inseguro desde 1998, época en la que una máquina diseñada expresamente para romper este sistema de cifrado (DESCracker) era capaz de probar todas las claves de DES

en nueve días. La longitud de cada bloque para DES es de 64 bits, esto es, equivalente a ocho símbolos ASCII. Su clave también tiene 64 bits, si bien el octavo de cada ocho se emplea para comprobar la paridad de los siete bits que le preceden y evitar errores en la clave. Así pues, las claves efectivas de DES tienen 56 bits. El número de posibles claves de este tamaño es $2^{56} \simeq 7{,}2 \cdot 10^{16}$, lo que es una fragilidad, dado que actualmente un criptosistema simétrico se considera seguro si su clave tiene al menos 80 bits.

A continuación se presenta, *grosso modo*, el funcionamiento de este sistema de cifrado, cuya importancia histórica es fundamental para entender el desarrollo de los sistemas de cifrado simétricos actuales. Su esquema se muestra en la figura 17.

FIGURA 17
Esquema de DES.



FUENTE: ELABORACIÓN PROPIA.

Para cifrar un bloque mediante DES, en primer lugar se ejecuta una permutación fija (y conocida) cuyo objetivo es llevar a cabo una difusión de sus bits para evitar, como ya se ha dicho, estructuras rígidas, pero no tiene repercusión criptográfica. A continuación, el bloque se divide en dos mitades de 32 bits cada una; la izquierda (I_0) y la derecha (D_0), de modo que las acciones que se van a llevar a cabo se iteran 16 veces y en cada iteración se utiliza una subclave de 48

bits diferente, k_r. Estas 16 subclaves se obtienen a partir de la clave original de 56 bits, k, mediante un proceso conocido que consiste en elegir determinados bits de la clave inicial y en determinado orden.

En cada iteración, el bloque derecho pasa a ser el bloque izquierdo de la siguiente iteración, es decir, $I_{r+1} = D_r$, mientras que el nuevo bloque derecho se logra mediante la suma, bit a bit, del bloque izquierdo con una modificación del bloque derecho que depende de una función no lineal y de la subclave de la iteración, esto es, $D_{r+1} = I_r + f(k_r, D_r)$.

Una vez concluida la decimosexta iteración, se concatenan las dos mitades de los bloques y se obtiene el bloque cifrado sin más que llevar a cabo la permutación inversa a la permutación inicial.

La función no lineal empleada en cada iteración utiliza como entradas la parte derecha del bloque y la subclave correspondiente y consiste en varios pasos. El primero es expandir el bloque desde 32 hasta 48 bits, repitiendo determinados bits del bloque en posiciones ya establecidas de antemano. A continuación se suman, bit a bit, los 48 bits de esta expansión con los de la subclave. Los 48 bits obtenidos se separan en ocho grupos de seis bits, de modo que cada grupo entra en una caja *S* diferente, que proporciona como salida cuatro bits. Los ocho grupos de cuatro bits resultantes se concatenan, dando lugar a un bloque de 32 bits que, como ya hemos dicho, se suma, bit a bit, con la mitad izquierda del bloque de la iteración considerada y se convierte en el bloque derecho de la siguiente iteración.

Dado que los sistemas de cifrado que se van a presentar a partir de este momento hacen uso de herramientas informáticas que codifican los mensajes según su contenido exacto, estos se cifrarán considerando su grafía correcta (con tildes, mayúsculas, etc.). Dicho de otro modo, como los criptogramas de los mensajes "Criptografía" y "criptografia" son completamente diferentes aunque se emplee el mismo sistema de cifrado y la misma clave, se considerarán mensajes escritos correctamente.

Si se desea cifrar mediante DES el siguiente mensaje de 64 bits (un bloque): "Ejemplos:" utilizando como clave (escrita en hexadecimal, es decir, empleando las cifras 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) 0000CAFECAFE1111 se obtiene el siguiente mensaje cifrado: ™Sø"_"@v.

Ahora bien, es muy probable que el cifrado de un mensaje proporcione caracteres ASCII no imprimibles (retorno de carro, retroceso, etc.), en cuyo caso, las implementaciones de DES suelen mostrar el resultado del cifrado mediante código hexadecimal. Así, cifrar la frase "Este es un ejemplo con el sistema de cifrado DES con la misma clave de antes" proporciona como resultado el siguiente criptograma (cada grupo de dos caracteres en hexadecimal equivale a 8 bits, esto es, al cifrado correspondiente a cada letra, considerada en su bloque correspondiente): D8 8D 67 10 56 E9 B2 C9 65 88 96 A6 25 D0 F1 B8 23 A4 D7 06 6C F8 34 5D 8C 19 43 C0 F1 A1 9E 3F 50 97 7D 40 8E 28 89 D9 F2 05 CE CE 16 B9 43 8C.

Con el fin de utilizar la arquitectura desarrollada para el DES y a la vez evitar la fragilidad que supone la longitud de sus claves, se ha recomendado el uso del llamado cifrador múltiple TripleDES o TripleDEA. Este sistema de cifrado lleva a cabo tres cifrados consecutivos de DES. En el primero se cifra la información mediante un DES con una clave, sea k_1; el segundo consiste en descifrar mediante DES el resultado del cifrado anterior con una segunda clave diferente, k_2, y, finalmente, se cifra el resultado obtenido en el segundo paso, pero ahora se puede volver a utilizar la primera clave, k_1, o una clave diferente, k_3. Así pues, la longitud de los bloques del TripleDES sigue siendo de 64 bits, pero puede utilizar claves de 128 bits (k_1 y k_2) o de 192 bits (k_1, k_2 y k_3), lo que hace que aún sea considerado seguro y apto para el cifrado de datos.

No obstante, una vez probada la debilidad de DES, el NIST americano publicó una nueva norma en 2001 definiendo el nuevo sistema de cifrado en bloque estándar, conocido como AES (*Advanced Encryption Standard*). Este nuevo

sistema de cifrado fue elegido después de una competición internacional convocada precisamente para elegir un sustituto de DES. AES utiliza bloques de 128 bits de longitud, mientras que las longitudes de sus claves pueden ser de 128, 192 o 256 bits.

Al margen de las diferencias en cuanto a las longitudes de las claves, de los bloques y del tratamiento de estos como entes completos o divididos en mitades, existe una diferencia en el diseño de AES que lo hace muy diferente de DES. Ambos utilizan cajas *S*, pero mientras que las de AES están plenamente justificadas de modo matemático, las de DES nunca han sido del todo explicadas ni aclaradas, lo que ha llevado a suponer que DES pudiera tener puertas traseras que permitieran el descifrado de datos sin necesidad de conocer la clave.

De hecho, cada caja *S* de DES es una tabla de doble entrada con 4 filas y 16 columnas, de modo que en cada celda hay un número comprendido entre 0 y 15. Estas cajas, como ya se ha dicho, toman como entrada 6 bits y producen como salida cuatro bits. La forma de proceder es la siguiente (en la tabla 9 se muestra la caja S_5 de DES): el primer y último bit de los 6 de la entrada se concatenan y dan lugar a 2 bits, es decir, a un número entre 0 y 3 que determina una fila de la caja. En el caso de que los 6 bits de entrada fueran 110110, los 2 bits de los extremos a los que nos referimos serían 10, es decir, el número decimal 2, lo que indica la fila identificada como 2 (subrayado en la tabla).

TABLA 9
Caja S_5 de DES.

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	**5**	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Por su parte, los cuatro bits restantes definen un número entre 0 y 15, que determina una de las columnas de la caja *S*. En el caso mencionado, los bits serían 1011, que equivalen al número decimal 11, esto es, la columna identificada como 11 (subrayado en la tabla).

Entonces, la salida de la caja son los cuatro bits que definen el número contenido en la celda que está en el cruce de la fila y columna determinadas. En el ejemplo considerado, sería el número 5 (en negrita y subrayado en la tabla), que en binario es 0101. La razón de esta forma de definir las cajas *S* y la distribución de los valores de las celdas no ha sido explicada ni justificada, por lo que siempre se ha dudado de su seguridad.

Por su parte, la caja *S* de AES proporciona el mismo número de bits de salida que los que toma como entrada y es una permutación de los 256 números comprendidos entre 0 y 255 (00 y FF, en hexadecimal). Su construcción matemática se ha definido de modo que minimice tanto la correlación entre la entrada y la salida como la probabilidad de propagación de diferencias y, a la vez, maximice la complejidad de la expresión de la transformación (Fúster *et al.*, 2012, cap. 3).

CAPÍTULO 5

Criptografía de clave asimétrica

La criptografía de clave pública no se basa tanto en operaciones lógicas como lo hacen los cifradores en flujo y en bloque, sino que lo hace en operaciones matemáticas que están relacionadas con determinados problemas, supuestamente difíciles de resolver. Por este motivo, los mensajes suelen ser números con los que llevar a cabo operaciones y no tanto colecciones de bits, aunque tal consideración depende del criptosistema de que se trate.

En las siguientes secciones presentaremos los principales criptosistemas de clave pública que se emplean en la actualidad. Para cada uno de ellos trataremos el problema matemático en el que se basa su seguridad.

Protocolo de acuerdo de clave de Diffie-Hellman

Como ya hemos comentado en el apartado "Criptografía simétrica y asimétrica" del capítulo 3, hoy se considera que la criptografía asimétrica surgió en 1976 a raíz de la publicación del artículo de Diffie y Hellman (Diffie y Hellman, 1976) en el que se propone un protocolo de intercambio de información por el que las dos personas participantes en él acaban obteniendo una información común. El protocolo se

define de tal manera que ninguna otra persona podrá conocer la información compartida por quienes participan en el protocolo a pesar de que el canal de comunicación utilizado sea inseguro. Este protocolo recibe el nombre de sus autores, esto es, Protocolo de acuerdo (o intercambio) de clave de Diffie-Hellman o DHKA (*Diffie-Hellman Key Agreement*). Debe tenerse en cuenta que este protocolo no es un criptosistema en sí mismo, puesto que no se lleva a cabo ningún tipo de cifrado de información; lo que permite es que, al final del mismo, ambas partes puedan utilizar la información que acaban compartiendo como si fuera una semilla que luego les permite acordar una clave. Lo más importante es que la información resultante solo la conocerán ambas partes, aunque haya adversarios que puedan llegar a conocer la información intercambiada a lo largo del protocolo.

Como es tradicional en criptografía, supondremos que Alicia y Bernardo son los nombres de los dos usuarios que participan en el protocolo, de modo que los dos se ponen de acuerdo (públicamente) en un grupo cíclico finito multiplicativo, que podemos suponer de la forma $\mathbf{Z}_p^*$ con $n = p-1$ elementos, y seleccionan un generador de dicho grupo, sea $g \in \mathbf{Z}_p^*$.

El protocolo de Diffie y Hellman es el siguiente:

1. Alicia y Bernardo eligen de forma aleatoria e independiente sendos números naturales a y b, respectivamente, menores que n, y calculan g^a y g^b que, como son potencias del generador, pertenecen al grupo $\mathbf{Z}_p^*$. Cada uno envía el elemento calculado al otro por el canal inseguro.
2. Tanto Alicia como Bernardo, que han recibido g^b y g^a, respectivamente, pueden calcular $(g^b)^a = g^{b \cdot a}$ y $(g^a)^b = g^{a \cdot b}$ que, de nuevo, están en $\mathbf{Z}_p^*$ y coinciden.

Una vez acabado el protocolo, tanto Alicia como Bernardo conocen el elemento del grupo $g^{b \cdot a} = g^{a \cdot b}$, que es la información que comparten y que les puede servir para derivar a partir de ella una clave de cifrado para futuros usos.

Un criptoanalista o adversario, llamado Carlos, que intercepte las transmisiones anteriores entre Alicia y Bernardo, conocerá, además de $\mathbf{Z}_p^*$, g y n, que son públicos, los valores de g^a y g^b que se han intercambiado, pero deberá calcular $g^{a \cdot b}$ para llegar a conocer la misma información que comparten Alicia y Bernardo. Desde el punto de vista matemático, calcular, en poco tiempo, $g^{a \cdot b}$, conocidos $\mathbf{Z}_p^*$, g, n, g^a y g^b se conoce como *problema de Diffie-Hellman* o DHP (*Diffie-Hellman Problem*). Hoy en día, si los valores del protocolo se eligen adecuadamente, el DHP es considerado como un problema computacional muy difícil; esto es, su solución necesita de algunos miles de años de computación dado que no se sabe cómo obtener el valor buscado a partir de los valores conocidos en un tiempo razonablemente corto.

A modo de ejemplo podemos considerar el grupo cíclico finito $\mathbf{Z}_{17}^*$, que tienen $n = 16$ elementos, y un generador suyo, sea $g = 6$ (otros posibles generadores de este grupo son: 3, 5, 7, 10, 11, 12 y 14). Siguiendo el protocolo anterior, tendríamos lo siguiente:

1. Alicia y Bernardo eligen al azar $a = 5$ y $b = 10$, calculan $g^a \equiv 6^5 \pmod{17} \equiv 7$ y $g^b \equiv 6^{10} \pmod{17} \equiv 15$ y se intercambian esos valores.
2. Alicia calcula $(g^b)^a \equiv 15^5 \pmod{17} \equiv 2$ y Bernardo calcula $(g^a)^b \equiv 7^{10} \pmod{17} \equiv 2$.

Al finalizar, ambos comparten el valor $g^{a \cdot b} \equiv 2 \pmod{17}$.

Recordemos que el cálculo de las potencias anteriores se puede hacer de modo eficiente solo con utilizar el algoritmo de elevar al cuadrado y multiplicar (apartado "Algunos conceptos matemáticos" del capítulo 3).

Por su parte, Carlos conocería $\mathbf{Z}_{17}^*$, $n = 16$, $g = 6$, $g^a = 7$, $g^b = 15$ y debería calcular $g^{a \cdot b}$.

A la vista de estos datos, bastaría con poder calcular el valor de a (o el de b) de la expresión $g^a = 7$ (de $g^b = 15$) y luego utilizar la otra potencia, $g^b = 15$ (o $g^a = 7$) para calcular $g^{b \cdot a}$ (o $g^{a \cdot b}$).

Ahora bien, el problema de calcular a conociendo g y g^a es, en definitiva, el cálculo de logaritmos. En efecto, recordemos que se define el *logaritmo* en base n del número z como la potencia a la que hay que elevar la base para obtener dicho número, esto es: $\log_n z = x$, precisamente si $n^x = z$. Así, $\log_{10} 1000 = 3$ porque $10^3 = 1000$ y $\log_2 1024 = 10$ porque $2^{10} = 1024$.

Este problema es, casi, el problema que queremos resolver, dado que en nuestro caso tenemos que calcular $\log_g g^a$ en el grupo $\mathbf{Z}_p^*$.

A diferencia de los logaritmos tradicionales (sobre los números reales), en el caso que nos ocupa se trabaja en un conjunto finito de elementos, del tipo $\mathbf{Z}_p^*$, de modo que todas las operaciones, incluida la exponenciación, se hacen módulo p. Por esta razón al cálculo de estos logaritmos se le ha dado en llamar *problema del logaritmo discreto* o DLP (*Discrete Logarithm Problem*).

De forma más concreta, se define el logaritmo discreto en base n del número z como la potencia a la que hay que elevar la base para obtener dicho número en el grupo finito considerado. Así, si el grupo es $\mathbf{Z}_p^*$, por ejemplo, se tiene que $\log_n z = x$, precisamente si $n^x \equiv z \pmod{p}$.

De hecho, para determinar que $a = 5$, en el ejemplo anterior en el que el grupo era $\mathbf{Z}_{17}^*$ y el generador era $g = 6$, bastaría con ir calculando las potencias del generador (haciendo módulo, es decir, dividiendo por 17) hasta llegar a la potencia deseada: $6^1 \pmod{17} \equiv 6$, $6^2 \pmod{17} \equiv 2$, $6^3 \pmod{17} \equiv 12$, $6^4 \pmod{17} \equiv 4$, $6^5 \pmod{17} \equiv 7$.

La cuestión es que esta forma de proceder solo es eficiente si el número de potencias (y divisiones) a realizar es pequeño. Donde realmente reside la dificultad es que en la práctica, el número p se elige de modo que sea un número primo muy grande (digamos de entre 500 y 600 dígitos) y entonces el número de cómputos que hay que realizar es enorme.

Se puede apreciar la dificultad de este problema sin necesidad de llegar a números tan grandes como los utilizados

en las aplicaciones reales. Baste, por ejemplo, con considerar el generador $g = 3$, el primo

p = 1234567890 1234567890 1234567890 1234567890 1234568119

e intentar calcular el exponente al que elevar 3 para obtener

2798407287 7641343374 7931545757 5857674132 632547395

Nótese que en este caso p solo tiene 50 dígitos y los números se han escrito separando sus dígitos en grupos de 10, como es tradicional en criptografía, para facilitar contar cuántos tiene (para el lector interesado, el logaritmo buscado es 1357924680 1357924680 1357924680 1357924680).

En definitiva, si p tiene más de 1024 bits, es decir, si es mayor que 2^{1023}, el DLP es, hoy por hoy, un problema computacionalmente muy difícil de resolver, por lo que los protocolos criptográficos cuya seguridad se basa en este problema se consideran seguros.

Criptosistema RSA

El criptosistema RSA, llamado así en honor a los tres investigadores que lo propusieron (Rivest *et al.*, 1978), Ronald Rivest (1947-), Adi Shamir (1952-) y Leonard Adleman (1945-), es uno de los sistemas de cifrado asimétrico más utilizado en la actualidad y consta, como es habitual en la clave asimétrica, de tres procedimientos: generación de las claves, cifrado y descifrado (Durán *et al.*, 2005; Fúster *et al.*, 2012, cap. 7).

La generación de las claves es un proceso relativamente sencillo y eficiente, dado que las operaciones que se deben ejecutar son fáciles de implementar y rápidas de ejecución. Este *protocolo de generación de las claves* lo puede llevar a cabo cada usuario. Así, Alicia por su cuenta ejecuta los siguientes pasos:

1. Genera dos números primos grandes p y q (para una seguridad a corto-medio plazo, el número de bits de cada uno de ellos debe estar comprendido entre 1024 y 2048, longitudes menores no se consideran recomendables). A continuación calcula $n = p \cdot q$ (cuyo número de bits será la suma de los correspondientes a p y q, esto es, entre 2048 y 4096 bits) y determina el valor del indicador de Euler, que es el número de elementos del grupo $\mathbf{Z}_n^*$, $\phi(n) = (p-1)(q-1)$.
2. Luego elige un número entero positivo, A, mayor que 1 y menor que $\phi(n)$, que sea primo con este, es decir, de modo que $\text{mcd}(A, \phi(n)) = 1$.
3. Finalmente, calcula el inverso del número A módulo $\phi(n)$, es decir, el único entero a, que cumple: $A \cdot a \equiv 1 \pmod{\phi(n)}$.

La clave pública de Alicia es el par de números (n, A) y su clave privada es a. Además, por cuestiones de seguridad, es decir, para proteger la clave privada, los números p, q y $\phi(n)$ se mantienen en secreto. El *módulo RSA* es n, A es el *exponente de cifrado* y a el *exponente de descifrado.*

Ahora, si otro usuario, por ejemplo Bernardo, desea cifrar y enviar un mensaje M a Alicia, seguirá el siguiente *protocolo de cifrado*. Lo primero que debe hacer es pedir a Alicia su clave pública (n, A). A continuación, codifica el mensaje M para transformarlo en un elemento de $\mathbf{Z}_n^*$, sea m, es decir, en un número comprendido entre 1 y $n-1$. Finalmente, Bernardo calcula el valor del mensaje cifrado y se lo envía a Alicia. Esta acción se ejecuta elevando el mensaje a la clave pública de Alicia y haciendo módulo n: $c \equiv m^A \pmod{n}$.

Una vez que Alicia recibe le mensaje cifrado, c, el *protocolo de descifrado* consiste en calcular $c^a \pmod{n} = m$, es decir, se trata de elevar el criptograma recibido a su clave privada y hacer módulo n. Teniendo en cuenta que como A y a son inversos módulo $\phi(n)$, se cumple: $c^a \pmod{n} \equiv (m^A)^a \pmod{n} \equiv m^{A \cdot a} \pmod{n} \equiv m$.

Para terminar el proceso solo queda que Alicia descodifique m para obtener M.

La implementación de los protocolos de cifrado y descifrado es, básicamente, la misma, dado que en ambos casos se llevan a cabo las mismas operaciones matemáticas, esto es, elevar un número a un exponente y hacer módulo otro número. La facilidad que supone su implementación y su seguridad son los hechos que más han influido para que sea uno de los criptosistemas más extendidos.

La eficiencia del algoritmo de cifrado se basa en la rapidez con que se pueden calcular las potencias anteriores utilizando el algoritmo de elevar al cuadrado y multiplicar (véase "Algunos conceptos matemáticos" en el capítulo 3).

A modo de ejemplo, supongamos que los dos primos que elige Alicia tienen 60 bits cada uno y son:

p = 68244346 4951956207 y q = 99142144 7738173759

por lo que el módulo RSA es

$n = p \cdot q$ = 676589 0880221240 6605314179 6724572113

Además, calcula

$\phi(n) = (p-1)(q-1)$ = 676589 0880221240 6437927688 4034442148

y selecciona como exponente de cifrado A = 65537, con lo que su exponente de descifrado es

a = 427961 6713439603 7299202807 4992199289

Es frecuente considerar $65537 = 2^{16}+1$ como exponente de cifrado. Ello se debe a que es primo y a que su expresión en forma binaria es 10000000000000001, lo que, con los algoritmos empleados hoy en día para la exponenciación modular, facilita mucho el cifrado de los mensajes (Durán *et al.*, 2005, ap. A). Otro valor recomendado para A es 3, debido

a que, además de ser primo, elevar al cubo es una operación muy rápida.

Una vez que Alicia ha calculado sus claves, hace público el par (n, A) para poder recibir mensajes cifrados.

Si Bernardo quiere cifrar y enviar a Alicia el mensaje "Ejemplo RSA", lo que debe hacer es, además de conseguir la clave pública de Alicia, codificar el mensaje como un número menor que n. Si esto no fuera posible, trocearía el mensaje de modo que cada parte fuera próxima a n, pero menor. Una forma de realizar este proceso consiste en cambiar cada una de las letras del mensaje por su equivalente expresión decimal (con tres dígitos) en el código ASCII y luego concatenarlas todas.

Si, por ejemplo, el mensaje fuera "RSA", su codificación daría lugar a la expresión: "082083065", dado que la R es el 082, la S el 083 y la A el 065. Ahora bien, si el mensaje es "Ejemplo", la codificación sería "069106101109112108111".

El hecho de que el tamaño del número a cifrar dependa de la longitud del mensaje no es una buena práctica, dado que puede suponer alguna debilidad. Por ello, se procede a realizar lo que se denomina un *relleno* (*padding*), esto es, completar el mensaje con espacios (032) hasta lograr el número más próximo posible y menor que n. En este caso, como el mensaje tiene 11 letras, el relleno se limita a solo un espacio:

$$m = 069106101109112108111032082083065032$$

Una vez que Bernardo ha codificado el mensaje de texto, procede a cifrarlo tal y como se ha dicho, es decir, calcula el criptograma:

$$069106101109112108111032082083065032^{65537} \pmod{676589088022124066053141796724572113} \equiv 616317776451266267488044810173932884$$

Este valor, recibido por Alicia, es el que ella debe elevar a su clave privada para conocer el mensaje original:

$$616317776451266267488044810173932884^{427961671343960372992028074992199289} \pmod{676589088022124066053141796724572113} \equiv 069106101109112108111032082083065032$$

Debe tenerse en cuenta que los valores numéricos que se han presentado anteriormente son a título de ejemplo, dado que la longitud de n es de solo 120 bits (cada primo tiene una longitud de 60 bits). Una clave real, similar a cualquiera de las claves personales que vienen dentro del chip de un DNIe, podría ser el módulo RSA de 2048 bits (producto de dos primos de 1024 bits cada uno) siguiente:

1176094728 5600722187 0076803888 2216935291 4722465775
6631500258 7321972662 0452322667 6981320032 0333851584
9237082308 3493041769 5262931352 8265414313 1287304881
8382753071 2411157595 6458034192 9640821865 1422365099
6616233184 5718602402 9604386968 3444315584 8909006398
9728309774 7793472313 2761501660 3461436632 4611518639
6230161236 0958863821 8153403128 8941009132 6699571030
3528545080 9455516207 9295364678 6466852901 6282008279
5753995603 7580639279 6137271002 1863900935 0359388967
8013541327 0023549296 7825627397 4821323024 0956398866
6393509957 9797566816 3553986829 8815126984 3166119132
8176839894 3598417331 6608956985 6937344296 2036541469
2184760035 3676479

que tiene 617 dígitos y cuyos factores primos son, respectivamente,

1204328148 3372916036 4292423276 9208462720 0787868925
1114536602 3000167100 2472607693 6509435386 4908395316
6853517249 3463677156 6927416779 3296290868 7821319127
4539205544 6047285494 4102132046 3964585383 4502557271
9153741961 6234631259 8073834901 6071204944 6121554324
3875002652 7989799071 0524759350 3935268174 4845614576
741318513

y

9765567052 3337119705 3806474899 4959407795 5257512722
4644785663 5310016228 4997321739 6341176103 0147684799
5496555694 2013620566 0856648743 5523681959 2666721831
5376749027 1336862432 9399380138 8151602598 8925990944
7468166353 0043789705 2673740454 4589802041 3170561112
0063772286 8059692840 7807274431 1901318215 5487300865
47822383

de 309 y 308 dígitos cada uno.

Como se puede apreciar por el tamaño de los números anteriormente presentados, la generación de claves del criptosistema RSA presenta una dificultad: la de determinar números primos grandes (de más de 300 dígitos). Este problema matemático se conoce como *problema de la primalidad* y, en general, requiere bastante tiempo de computación. La estrategia consiste en generar un número impar al azar y luego llevar a cabo un test para analizar si tal número cumple las propiedades que verifican los números primos. Estas pruebas son probabilísticas, por lo que si la salida de la misma es: "el número no es primo", se puede asegurar que, en efecto, no lo es. Sin embargo, si la salida es: "el número es primo", lo que se puede afirmar es que probablemente el número sea primo, con una alta probabilidad que puede elegirse de antemano y que suele ser del orden del 99,9999%. La prueba que se suele emplear se conoce como Test de Miller-Rabin (Durán *et al.*, 2005, cap. 5) y es la que requiere menos tiempo de computación. Existen algoritmos que verifican la primalidad de un número de modo determinista, pero no son prácticos porque, incluso los más rápidos, como el llamado AKS, consumen mucho tiempo de computación (Agrawal *et al.*, 2004).

El otro problema matemático relacionado con el RSA es el que tiene que ver con su seguridad. Dicho de otra manera, se debe garantizar, de algún modo, que el hecho de conocer la clave pública, la forma de generarla y los protocolos de cifrado y descifrado no supone ningún perjuicio

para el propietario de las claves ni una ventaja para un atacante al sistema.

El problema matemático que garantiza la seguridad del RSA es el *problema de la factorización de números enteros*, que consiste en calcular la descomposición en factores de un número compuesto, esto es, de un número no primo. Este problema se considera, hoy en día, como uno de los más difíciles de resolver si el número a factorizar se elige convenientemente. Muchos números son fáciles de factorizar, basta con que sean pequeños o tengan muchos factores primos. Sin embargo, si el número a factorizar es grande y tiene muy pocos factores, la complejidad computacional es muy elevada.

Que el problema de la factorización es la base de la seguridad del RSA se deduce del hecho de que si fuera fácil, desde el punto de vista computacional, determinar los factores primos que dividen a un número compuesto, entonces sería fácil calcular los valores de p y q en la expresión del módulo RSA, n. Hecho esto, como el valor de A es público y $\phi(n)$ sería calculable fácilmente, entonces a se podría determinar por medio del algoritmo de Euclides y el sistema quedaría vulnerado (Durán *et al.*, 2005, cap. 6).

En todo caso, se han propuesto diferentes ataques para romper el RSA, pero todos ellos resultan infructuosos si se toman las medidas oportunas para evitarlos (Durán *et al.*, 2005, cap. 7).

Criptosistema de ElGamal

El criptosistema de ElGamal fue publicado por Taher ElGamal (1955-) (ElGamal, 1985) y, además de ser otro de los criptosistemas de clave asimétrica más extendidos, es el que ha dado lugar a los criptosistemas con mayor futuro, los basados en curvas elípticas, que veremos posteriormente.

Al igual que el sistema RSA, este criptosistema necesita generar unas claves y luego definir los procesos de cifrado y descifrado. El *protocolo de generación de claves* es el siguiente:

1. Alicia genera un número primo grande p (de entre 2048 y 4096 bits) y elige un generador, g, del grupo $\mathbf{Z}_p^*$.
2. A continuación, genera un número aleatorio a, mayor que 1 y menor que $p-2$, y calcula $g^a \pmod p \equiv A$.

La clave pública de Alicia será (p, g, A); mientras que a es su clave privada. Para que Bernardo cifre un mensaje M para Alicia, una vez que conoce su clave, debe proceder como sigue:

1. En primer lugar ha de codificar M como un elemento de $\mathbf{Z}_p^*$, es decir, como un número, m, comprendido entre 1 y $p-1$.
2. A continuación debe generar un número aleatorio x, mayor que 1 y menor que $p-2$, y calcular $r \equiv g^x \pmod p$ y $s \equiv m \cdot A^x \pmod p$, de modo que el texto cifrado es $c = (r, s)$.

Para que el destinatario, Alicia, pueda descifrar el criptograma y recuperar el mensaje original, debe seguir el siguiente *protocolo de descifrado*:

1. Alicia utiliza su clave privada, a, para calcular $t \equiv r^{p-1-a} \pmod p$.
2. A continuación, multiplica el valor anterior por s y obtiene m.

En efecto, el valor de t es, en realidad, el siguiente:

$$t \equiv r^{p-1-a} \pmod p \equiv (g^x)^{p-1-a} \pmod p) \equiv (g^{(p-1)})^x \, g^{-x \cdot a} \pmod p \equiv g^{-x \cdot a} \pmod p$$

y multiplicarlo por s consiste en calcular

$$t \cdot s \pmod p \equiv g^{-x \cdot a}(m \cdot A^x) \pmod p \equiv g^{-x \cdot a} \cdot m \cdot g^{a \cdot x} \pmod p \equiv m$$

El hecho de que las operaciones para el cifrado y el descifrado no sean análogas (como sí sucede en el caso del RSA) y que el criptograma tenga un tamaño doble que el del mensaje a cifrar (en el RSA tenía el mismo tamaño) ha hecho que este criptosistema no haya adquirido tanta popularidad como el RSA.

Como ejemplo, supongamos que se elige un número primo de 120 bits, sea

p = 759299 1330708747 7716462624 7483495829

y un generador del grupo, $g = 2$ (otros generadores son 3, 7, 8, 10…, 23123129,…). Como número aleatorio, Alicia selecciona su clave privada

a = 115877 5384527247 2305268712 1128149276,

con lo que su clave pública, además de los valores de p y g, es

A = 193517 2106552475 9799711323 0307849315.

Alicia hace pública su clave (p, g, A) para que sus amigos le puedan enviar mensajes cifrados. Así, si Bernardo quiere cifrar y enviar a Alicia el mensaje "Ej. ElGamal" deberá codificarlo y luego cifrarlo con el protocolo anterior. El mismo tipo de codificación que se hizo con el RSA proporciona el siguiente mensaje

m = 06910604603206910807109710909710 8032.

Ahora, para que Bernardo cifre el mensaje anterior para Alicia, deberá elegir al azar un número mayor que 1 y menor que p–2, por ejemplo,

x = 263961346 2155756083 7970494629 8386565493

y calcular r y s, como sigue:

$r \equiv g^x \pmod{p} = 2^{263961346215575608379704946298386565493}$
$\pmod{75929913307087477716462624748349582 9} \equiv$
740626969274717311543545109924726289,
$s \equiv m \cdot A^x \pmod{p} \equiv 069106046032069108071097109097108032 \cdot$
$193517210655247597997113230307849315^{263961346215575608379704946298386565493}$
$\pmod{75929913307087477716462624748349582 9} \equiv$
619167502131428903574710522465875275,

de modo que el texto cifrado será el par dado por los dos últimos números *r* y *s*:

(740626969274717311543545109924726289,
619167502131428903574710522465875275)

Para que Alicia descifre el mensaje original, deberá utilizar su clave privada, calcular

$t \equiv r^{p-1-a} \pmod{p} \equiv$
$643421594618150054111939126355346552^{643421594618150054111939126355346552}$
$\pmod{75929913307087477716462624748349582 9} \equiv$
165402685638191266703216847237494124

y luego obtener:

$t \cdot s \pmod{p} \equiv 165402685638191266703216847237494124 \cdot$
619167502131428903574710522465875275
$\pmod{75929913307087477716462624748349582 9} \equiv$
691060460320691080710971090971080 32,

que es el mensaje original.

De nuevo, debe tenerse en cuenta que los tamaños de los números anteriores no se corresponden con los tamaños utilizados en la práctica.

Con relación a la seguridad de este sistema de cifrado, se puede observar que un adversario podría conocer la clave pública de Alicia y eventualmente los valores que le ha enviado

Bernardo; es decir, puede conocer p, g, $A = g^a$, $r = g^x \pmod p$ y $s = m \cdot g^{a \cdot x} \pmod p$. Pero para determinar la clave privada de Alicia, a, o calcular el valor de x, tendría que resolver un caso del problema del logaritmo discreto, lo cual es computacionalmente muy difícil. Por esta razón, se afirma que la seguridad del criptosistema de ElGamal se basa en el problema del logaritmo discreto (Fúster *et al.*, 2012, cap. 7).

Criptosistema de curvas elípticas

Los criptosistemas anteriores se consideran seguros si los tamaños de las claves utilizadas son grandes, es decir, de unos 2048 bits. No obstante, este tamaño hace que sea complicado implementar sistemas de cifrado en dispositivos con poca capacidad de cálculo o con poco espacio físico, como es el caso de las tarjetas inteligentes (bancarias, de identificación, telefónicas, etc.). Por este motivo, se han propuesto grupos alternativos a $\mathbf{Z}_n^*$ y a $\mathbf{Z}_p^*$ para usarlos como base e implementar con ellos sistemas de cifrado que requieran menor capacidad de cómputo, pero que ofrezcan niveles de seguridad equivalentes.

El conjunto más empleado como alternativa a los mencionados anteriormente es el grupo que forman los puntos de un tipo especial de curvas, llamadas curvas elípticas (Fúster *et al.*, 2012, cap. 8), y que ha dado lugar a lo que se conoce como *Criptografía basada en curvas elípticas* o ECC (*Elliptic Curve Cryptography*). Estas curvas suelen representarse mediante la ecuación conocida como de Weierstrass, cuya expresión es de la siguiente forma: $y^2 + a_1\, x\, y + a_3\, y = x^3 + a_2\, x^2 + a_4\, x + a_6$.

Si la curva se considera definida sobre el plano, tiene infinitos puntos, dado que hay infinitos pares (x, y) que cumplen la anterior ecuación. Sin embargo, estas curvas tienen un conjunto finito de puntos cuando se definen sobre un cuerpo finito, como por ejemplo $\mathbf{Z}_p$, de modo que las coordenadas de los mismos son elementos de $\mathbf{Z}_p$. Además, es posible definir la de suma dichos puntos, de modo que esta operación hace que

el conjunto de puntos de la curva sea un grupo cíclico aditivo. Dicho de otro modo, en estas curvas se puede seleccionar un generador, de modo que los restantes puntos son múltiplos del generador.

Considerando la estructura definida por los puntos de una de tales curvas, es posible definir un sistema de cifrado asimétrico de forma análoga a como se ha hecho con el sistema de ElGamal. Sin embargo, existen varias diferencias importantes entre ambos tipos de sistemas.

La primera de ellas es que la longitud de las claves es muy diferente para una seguridad equivalente. Así, una curva elíptica cuyas claves sean de 160 bits proporciona la misma seguridad que si se emplearan 1024 en el sistema de ElGamal (o en el RSA); análogamente, curvas elípticas con claves de 224 bits proporcionan una seguridad equivalente a claves de 2048 bits en ElGamal (o en RSA). Por tanto, la reducción en el tamaño de la clave es sustancial.

Por contraposición, la operación de suma que se lleva a cabo con los puntos de una curva elíptica tiene una mayor complejidad que en el caso del producto en el sistema de ElGamal.

Con relación a la seguridad, se puede decir que si esta depende de la dificultad de resolver el logaritmo discreto, como sucede con ElGamal, en el caso de las curvas elípticas esta dificultad se basa en la dificultad de resolver el problema del *logaritmo elíptico*. Se puede decir que este problema es la traducción al caso aditivo del logaritmo discreto utilizado en el caso multiplicativo.

Dicho de otra manera, el logaritmo elíptico se puede definir como sigue: dado el grupo de puntos de una curva elíptica, E, sobre un cuerpo finito, generado por el punto G y dado un punto de la curva $P \in E$, se trata de determinar el número x tal que $x \cdot G = P$.

En todo caso, la ventaja del utilizar curvas elípticas como grupo base para definir un criptosistema radica, fundamentalmente, en que es posible llevar a cabo implementaciones eficientes en dispositivos de poca capacidad física, como es el caso de las tarjetas inteligentes.

CAPÍTULO 6

Usos actuales y tendencias futuras

En esta sección vamos a presentar algunas primitivas criptográficas y las aplicaciones más importantes de la criptografía en la sociedad actual. También veremos algunas de las líneas presentes y futuras en las que se está trabajando y que tienen un gran porvenir.

Las aplicaciones para el cifrado de datos, las que generan firmas electrónicas o, en general, las que hacen uso de otras primitivas criptográficas han sido desarrolladas por especialistas en estos temas y su uso se ha generalizado tanto que sus implementaciones suelen ser transparentes para el usuario. Es decir, el usuario se limita a dejar que la aplicación proceda según el objetivo que se pretenda y en este proceso el usuario solo da su consentimiento a la aplicación para que esta haga uso de sus datos confidenciales, aportando alguna contraseña o frase de acceso.

Esquema de cifrado híbrido

El primero de los protocolos criptográficos que presentamos es el que se conoce como *esquema de cifrado híbrido* (Fúster *et al.*, 2012, cap. 7). Este tipo de esquema utiliza las ventajas de los criptosistemas simétricos y asimétricos y trata de evitar

sus inconvenientes. Así, los esquemas híbridos se aprovechan de la velocidad de los primeros y de la longevidad de las claves de los segundos para el cifrado y descifrado de mensajes.

En este caso, dos usuarios desean intercambiarse mensajes cifrados y ambos poseen sus respectivas claves públicas y privadas, así como la clave pública del otro. Además, ambos se ponen de acuerdo en el uso de un criptosistema de clave simétrica, pero no comparten la clave secreta del mismo porque no están cerca, hace mucho que no se ven o no se van a ver en un largo tiempo.

Si Alicia desea cifrar un mensaje a Bernardo lleva a cabo el siguiente protocolo híbrido de cifrado:

1. Alicia elige aleatoriamente una clave (de sesión) k para el criptosistema simétrico acordado con Bernardo.
2. Alicia cifra el mensaje a enviar, por ejemplo m, haciendo uso de tal criptosistema simétrico, obteniendo el criptograma c.
3. Alicia cifra la clave secreta k que ha generado, mediante el criptosistema de clave pública elegido, usando la clave pública de Bernardo, obteniendo K.
4. Alicia envía a Bernardo el par formado por (c, K).

Para que Bernardo obtenga el mensaje original, hace lo siguiente:

1. En primer lugar obtiene la clave secreta, k, sin más que descifrar K mediante su clave privada.
2. Bernardo descifra el criptograma c y obtiene el mensaje original, m, haciendo uso del criptosistema simétrico con la clave secreta obtenida k.

Como se puede ver, cada usuario solo conoce la clave pública del otro, mientras que la clave secreta que se emplea en cada sesión es recuperada por el destinatario, quien hace uso de su clave privada. La clave de sesión se desecha una vez terminada la misma y no vuelve a utilizarse.

Este tipo de esquema se emplea cuando un usuario se conecta con una página web segura. Tal hecho se señala mediante el indicativo *https://* (en lugar del tradicional *http://*) delante de la dirección web del lugar al que se realiza la conexión, lo que garantiza que la comunicación entre ambas partes es segura porque va siempre cifrada.

Funciones resumen

Entre las primitivas criptográficas de mayor uso en la actualidad destacan las funciones resumen (*hash*, en inglés). Estas funciones no llevan a cabo ningún proceso de cifrado ni descifrado, pero permiten comprobar, entre otras cosas, la integridad de determinada información. Las funciones resumen son funciones públicamente conocidas que transforman una información o mensaje de cualquier tamaño en una información que tiene un tamaño fijado de antemano (Fúster *et al.*, 2012, cap. 6).

Resulta claro que como los resúmenes para una función dada tienen siempre la misma longitud, sin depender de la longitud del mensaje, el número de todos los resúmenes diferentes que se pueden obtener es mucho menor que el de mensajes. Dicho de otro modo, siempre habrá mensajes diferentes cuyos resúmenes coincidan.

Así, si una función proporciona resúmenes de, por ejemplo, 256 bits, el número de tales resúmenes solo es de 2^{256}, mientras que el número de posibles mensajes de cualquier longitud es infinito.

A estas funciones se les exige verificar determinadas propiedades. Así, la primera de ellas es que sean *fácilmente computables*, es decir, el tiempo necesario para determinar el resumen de un documento debe ser muy pequeño. La segunda se denomina *dependencia de bits*, lo que significa que el resumen de un documento dependerá de todos los bits del mensaje, de tal manera que si se modifica un único bit del documento, el resumen debe cambiar, por término medio, en la mitad de sus bits.

Estas funciones se utilizan, entre otros, con los siguientes fines:

- Integridad de datos: para garantizar la integridad de los ficheros guardados en un dispositivo (computador, USB, etc.), enviados por correo electrónico o depositados en un registro o custodia, basta con calcular los resúmenes de los ficheros que interesen y guardar dicho valor en lugar no accesible o imprimirlo. Para comprobar si tales ficheros han sido modificados después de un tiempo, basta con volver a calcular sus resúmenes y comparar si los primeros y los últimos coindicen.
- Detección de software dañino: los programas antivirus intentan detectar la presencia de software dañino (*malware* en inglés) en un dispositivo calculando el resumen de los ficheros que son susceptibles de infectar el dispositivo. Cada uno de estos resúmenes es comparado con los que están en la base de datos que posee el programa y que es actualizada por la empresa que lo desarrolla. En caso afirmativo, el fichero analizado se considera dañino y el programa informa de ello. De ahí la importancia de que el programa antivirus que se posea esté siempre actualizado.

Otro de los usos más comunes de las funciones resumen está en las firmas electrónicas y en los certificados digitales, como se verá más adelante.

Con relación a las propiedades de seguridad de estas funciones, es importante que no sea fácil computacionalmente determinar dos mensajes diferentes que tengan el mismo resumen. Por ello, estas funciones deben cumplir las siguientes propiedades:

- Resistencia a la preimagen: conocido el resumen de un mensaje, debe ser computacionalmente difícil obtener un mensaje cuyo resumen sea el conocido; dicho

de otro modo: toda función resumen debe ser difícil de invertir.
- Resistencia a la segunda preimagen: dado un mensaje cualquiera debe ser computacionalmente difícil encontrar otro mensaje diferente cuyo resumen coincida con el del anterior.
- Resistencia a colisiones: debe ser computacionalmente difícil encontrar una colisión, esto es, determinar dos mensajes distintos cualesquiera cuyos resúmenes coincidan.

Es importante señalar que las dos últimas condiciones son diferentes, dado que en la primera de ellas se supone conocido uno de los mensajes, mientras que en la otra no se imponen condiciones sobre los mensajes. Así pues, la resistencia a colisiones es una condición más débil que la resistencia a la segunda preimagen.

Estas condiciones de seguridad son necesarias para evitar que las funciones resumen sean vulnerables. El principal ataque contra ellas se basa en la llamada *paradoja del cumpleaños*, que se puede enunciar de la siguiente manera: ¿cuántas personas debe haber en una sala para que la probabilidad de que dos de ellas celebren su cumpleaños el mismo día sea mayor del 50%? El hecho de que a este problema se le denomine paradoja se debe al sorprendente valor de su solución: basta con que en la sala haya 23 personas para que la probabilidad de que dos de ellas celebren su cumpleaños el mismo día sea mayor del 50%.

Firmas electrónicas o digitales

El protocolo de firma surgió de la necesidad de garantizar que los mensajes cifrados fueran realmente enviados por quien decía ser el remitente, dado que, si no se tomaban las oportunas medidas de seguridad, una persona podía suplantar a otra y enviar mensajes cifrados en su nombre. Hoy en día la firma electrónica es, probablemente, la aplicación o protocolo

criptográfico más utilizado en todos los ámbitos (Fúster *et al.*, 2012, cap. 9). Se debe tener en cuenta que los protocolos de firma requieren, para ser implementados, de un criptosistema asimétrico, de modo que el firmante debe poseer una clave pública y su correspondiente clave privada asociada. Actualmente, las firmas electrónicas más extendidas se llevan a cabo mediante el criptosistema RSA (véase "Criptosistema RSA" en el capítulo 5).

La firma electrónica o digital es un protocolo criptográfico que permite demostrar la autoría y autenticidad de un documento digital. Cada firma se determina utilizando el documento a firmar, una función resumen y la clave privada del firmante. Así pues, si se cambia el documento a firmar, la firma es diferente. De la misma forma, si dos usuarios distintos firman el mismo documento, también se obtendrán diferentes firmas. En definitiva, la firma de cada documento para cada firmante es única. En este sentido, la firma electrónica difiere sustancialmente de la manuscrita, dado que esta última suele ser siempre la misma, independientemente del documento que se firme.

La verificación de la firma la puede realizar cualquier otro usuario que conozca el mensaje, la firma y la clave pública del firmante.

La Ley 59/2003 (Cortes Generales, 2003) distingue tres tipos de firma: firma electrónica, firma electrónica avanzada y firma electrónica reconocida y otorga a la firma electrónica la misma validez legal que a la firma manuscrita.

Todo protocolo de firma electrónica consta de dos partes: en la primera parte el firmante elabora su firma para un documento y en la segunda cualquiera que lo desee puede verificar dicha firma.

Si Alicia desea firmar electrónicamente un documento no secreto, m, siendo su clave pública A y su clave privada a, lleva a cabo el siguiente protocolo:

1. Alicia calcula el resumen del documento a firmar m, mediante una función resumen públicamente conocida, obteniendo r.

2. A continuación Alicia cifra con su clave privada, *a*, el resumen del mensaje, *r*, obteniendo su firma para dicho mensaje: *s*.
3. Alicia hace públicos, tanto el mensaje como su firma para el mismo: (*m*, *s*).

Si Bernardo quiere verificar la validez de la firma de Alicia, *s*, para el mensaje *m*, procede como sigue:

1. Bernardo calcula el resumen del documento firmado, *m*, mediante la misma función resumen que utilizó Alicia y obtiene *r*.
2. Posteriormente, Bernardo descifra con la clave pública de Alicia (que conoce por ser pública) la firma del mensaje, *s*, obteniendo *R*.
3. Finalmente, Bernardo comprueba si los valores de *r* y de *R* coinciden. En caso afirmativo, la firma queda verificada. En caso contrario, la firma para el mensaje se rechaza.

De los protocolos anteriores se deduce que solo Alicia ha sido la firmante del mensaje, porque solo ella posee su clave privada y, además, que el mensaje no ha sido manipulado.

Existen otros tipos de firma, como las ciegas, las delegadas, las grupales, las múltiples, las autocertificadas, etc. Pero estas son aplicaciones diseñadas específicamente para resolver determinadas situaciones y no corresponden al planteamiento general y estándar de una firma electrónica clásica.

Certificados digitales

Si un certificado, emitido por una autoridad reconocida, permite a su poseedor probar ante terceros que posee determinado conocimiento o aptitud, un certificado digital (o electrónico) permite a su poseedor probar ante terceros que posee una clave criptográfica; es decir, se trata de la versión digital

de un certificado ordinario en el que se garantiza que la clave pública y el resto de información contenida en el mismo pertenecen al usuario que se especifique en dicho certificado (Fúster *el al.*, 2012, cap. 10). La validez de dicha información está garantizada por una entidad reconocida (local, nacional o internacional), a modo de notario electrónico, denominada autoridad de certificación o AC (Durán *et al.*, 2005).

Los certificados digitales más usados corresponden al estándar denominado X.509 (v.3) y contienen, entre otra, la siguiente información:

- Identificación del certificado mediante su número de serie.
- Versión del certificado.
- Identificador del algoritmo de firma digital que se emplea.
- Autoridad de certificación que emite el certificado y que garantiza su contenido.
- Identificación del usuario del certificado.
- Tipo de criptosistema de clave pública que emplea el usuario.
- Clave pública y privada del usuario.
- Periodo de validez del certificado que, en general, depende del tamaño de la clave (tradicionalmente varía entre 2 y 3 años).
- Firma digital de la autoridad que avala el certificado.

Algunos de los datos anteriores se pueden ver en el certificado digital que se muestra en la figura 18.

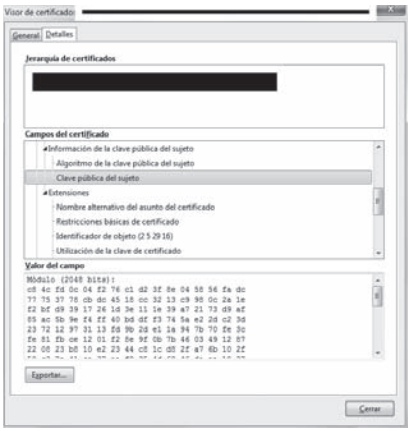
Para obtener un certificado, el usuario y la autoridad de certificación interactúan siguiendo un proceso similar al siguiente:

1. El usuario solicita a la autoridad la expedición de un certificado, en general, vía Internet.
2. La autoridad de certificación pide al usuario sus datos personales, comprobando la veracidad de los mismos.

3. La AC solicita al navegador del usuario que genere las claves pública y privada para dicho usuario.
4. La autoridad hace que se genere un fichero electrónico con los campos correspondientes al tipo de certificado solicitado.
5. La AC firma digitalmente el resumen del contenido del fichero generado y añade esta firma a dicho fichero, resultando de esta unión el certificado digital.

Para verificar la veracidad del contenido de un certificado basta con separar la firma de la AC de los restantes datos y comprobar que la firma del certificado es la firma del mensaje formado por los restantes datos.

FIGURA 18
Certificado digital.



Estos certificados permiten que su propietario firme electrónicamente documentos o ficheros, dado que sus claves están almacenadas en el mismo. Además, se pueden emplear en otros usos como los siguientes:

- Correo electrónico seguro: basta con que un usuario envíe su certificado digital a sus contactos y que estos

le cifren los mensajes haciendo uso de la clave pública almacenada en el mismo.

- Navegación segura: los navegadores de un usuario (cliente) y del lugar donde se conecta (servidor), como su banco, por ejemplo, pueden utilizar las claves necesarias para que toda la información que se intercambien esté cifrada y a salvo de posibles adversarios. Existen métodos para llevar a cabo este proceso como el protocolo de capa de conexión segura (*Secure Sockets Layer* o SSL, no recomendado desde 2014 por ser inseguro) o el protocolo de seguridad de la capa de transporte (*Transport Layer Security* o TLS).
- Comercio electrónico seguro: es posible, y cada día más fácil y seguro, comprar artículos a través de Internet, guardando ciertas precauciones elementales, por ejemplo verificar que se está realizando la compra a través de un canal seguro (como el ya comentado *https://*), el protocolo de transacción electrónica segura (*Secure Electronic Transaction* o SET) u otros similares.

DNIe

El Documento Nacional de Identidad electrónico ha venido a sustituir al clásico DNI para la identificación de su titular, pero ahora incluye un chip con capacidades criptográficas que permite llevar a cabo procesos de firma electrónica con la misma validez que la tradicional firma manuscrita (Fúster *et al.*, 2012, cap. 10). El actual DNIe[3] posee diferentes características de seguridad según se considere su soporte físico o lógico (Crespo *et al.*, 2006).

La principal información almacenada en el chip es la siguiente:

3.Véase http://dnielectronico.us/Asi_es_el_dni_electronico/presen_graf.html para un ejemplo del actual DNIe.

- Datos de filiación del ciudadano.
- Imagen digitalizada de la fotografía.
- Imagen digitalizada de la firma manuscrita.
- Plantilla de la impresión dactilar.
- Tres certificados: de autenticación, de firma (no repudio) y de la AC emisora.

Además, al chip del DNIe se le exigen unos elevados requisitos de seguridad para garantizar que pueda ser utilizado para la firma electrónica reconocida y avanzada (Cortes Generales, 2003), como los siguientes:

- Certificación del chip del DNIe según los criterios estándares internacionales ,81,*Common Criteria*, con un nivel de confianza de evaluación (*Evaluation Assurance Level*) equivalente a EAL 5+.
- Dispositivo seguro de creación de firma según el acuerdo del grupo de trabajo del Comité Europeo de Normalización.
- Las aplicaciones que se ejecuten en el DNIe tendrán el nivel de certificación *Common Criteria*, al menos, EAL 4+.

Como hemos dicho, el DNIe contiene dos certificados digitales (X.509) relacionados con su propietario:

- El de autenticación del ciudadano, que permite acreditar electrónicamente su identidad frente a terceras partes.
- El de firma o de no repudio del ciudadano, que garantiza que un documento firmado electrónicamente por el titular no ha sido alterado, de modo que acredita la procedencia del documento firmado y la identidad del firmante.

La versión más actual del DNIe, que empezó a emitirse a principios del año 2015, es la conocida como versión

3.0. Esta versión posee algunas características novedosas con relación a su versión anterior, como es el hecho de que está dotado de una tecnología denominada NFC (*Near Field Communication*) o comunicación de campo cercano y que permite simplificar su uso con teléfonos móviles y tabletas, evitando, en este caso, el uso de los lectores de tarjetas. Las características más importantes de esta tecnología son su velocidad casi instantánea, que puede enviar y recibir información de forma simultánea, y que no precisa de un emparejamiento previo (como sucede en el caso de la conexión por *bluetooth*). Su principal inconveniente es que el alcance es muy pequeño al estar limitado a unos 20 cm.

Existen numerosas aplicaciones que hacen uso del DNIe[4], como el acceso a bancos, la consulta del padrón municipal o de los puntos del carnet de conducir, la presentación de solicitudes en la Administración pública, etc.

Otros protocolos criptográficos

En esta sección se presentan, de forma muy resumida, las características más importantes de otros protocolos en los que se emplean primitivas criptográficas.

Uno de los protocolos que está teniendo cada día más auge es el *voto electrónico*. En este, se lleva a cabo una elección de modo que:

- Solo pueden ejercer su derecho al voto los votantes registrados en el censo.
- Cada votante solo puede votar una vez.
- Cada voto es secreto.
- Todos los votos han de ser contabilizados en el resultado.
- Cada votante puede verificar que su voto ha sido considerado en el recuento de la votación.

4. Véase http://www.dnielectronico.es/servicios_disponibles/index.html

Otro protocolo criptográfico se emplea en la *computación con datos cifrados*. En este caso, un usuario o empresa necesita hacer determinados cálculos con datos pero no dispone de la capacidad de cómputo necesaria y contrata a otra empresa que los ejecute en su lugar. Sin embargo, el usuario no desea que la empresa contratada tenga acceso a los datos en claro, para lo cual se los suministra cifrados. La empresa hace los cálculos con tales datos cifrados y se los devuelve al usuario, quien es capaz de interpretar los datos que corresponden a tales cómputos. Esta situación es frecuente en nuestros días cuando determinadas empresas contratan los servicios de otras para que les hagan *computación en la nube* (*cloud computing*, en inglés) o, sencillamente, les almacenen los datos cifrados, no en claro, para proteger los datos de sus clientes o como datos de respaldo a modo de copia de seguridad.

Uno de los principales problemas con los que se encuentran las empresas desarrolladoras de software es la copia y distribución ilegal de sus productos, entre ellos los juegos para ordenadores, móviles y consolas. Para evitar las pérdidas que esta práctica les acarrea, dichas empresas utilizan sistemas de *protección de software*. Una de tales protecciones (además de las protecciones físicas) consiste en firmar digitalmente sus productos, por ejemplo el DVD en el que se vende el software, de modo que cuando el dispositivo (Xbox, PlayStation, Wii, etc.) vaya a ejecutar el software, el primer paso que realiza es verificar la firma electrónica del DVD para determinar si este es una copia legal.

La *identificación amigo-enemigo* es otra de las grandes aplicaciones de la criptografía que se emplea para, por ejemplo, verificar la identidad de alguien que se declara amigo. En este caso, la aplicación plantea al usuario, cuya identidad trata de verificar, un protocolo de desafío-respuesta, de tal manera que si el reto es respondido satisfactoriamente, el usuario es identificado como amigo, mientras que, en caso contrario, se le declara enemigo. Normalmente, la forma de proceder consiste en que la aplicación, que posee todas las claves públicas de sus amigos, genera un mensaje aleatorio, lo cifra con la

clave pública del usuario a identificar y se lo envía. El mensaje descifrado por dicho usuario es devuelto al sistema y si este coincide con el mensaje original se le considera amigo, dado que es el único que posee la clave privada asociada a la clave pública utilizada por el sistema.

Una aplicación relacionada con la anterior es el *acceso a los servicios* de una red. En este caso, lo normal es que el usuario acceda mediante su nombre de usuario (*login*) y su contraseña (*password*). Sin embargo, con el fin de evitar que un atacante pudiera tener acceso a las contraseñas de los usuarios almacenadas en el sistema, lo que se suele hacer es guardar, junto a cada nombre de usuario, el resumen de su contraseña (calculado mediante una función resumen), de modo que para identificar a un usuario basta con comprobar que el resumen de la contraseña que ha introducido corresponde con el dato almacenado junto a su nombre de usuario.

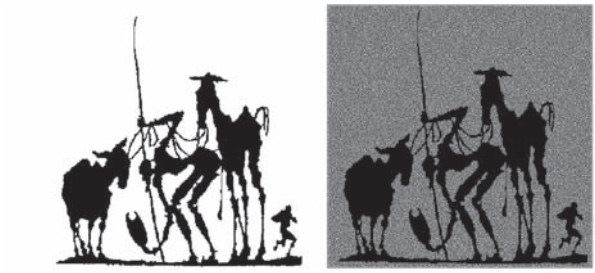
La aplicación de *reparto o compartición de secretos* se presenta cuando alguien posee solo una copia de un secreto y desea evitar perderlo, pero tampoco quiere tener varias copias del mismo para evitar que le sea robado. En este caso se reparte el secreto en varias partes o sombras (ninguna de las cuales aisladamente proporciona información sobre el secreto), de modo que para recuperar el secreto original se necesita el concurso de un mínimo número de ellos.

Para finalizar esta sección, comentaremos un tipo de criptografía que ha proliferado en los últimos años, fundamentalmente debido al auge de los canales multimedia, que hacen un uso continuo y frecuente de imágenes. Se trata de la *criptografía visual*. Esta forma de cifrado es una propuesta de reparto de secretos aplicada expresamente al caso de imágenes. Se trata de obtener, a partir de una imagen secreta, una serie de sombras que vuelven a ser imágenes y que se imprimen en transparencias de acetato. Posteriormente, la imagen secreta se recupera a partir de las sombras generadas sin más que superponer las transparencias adecuadas, es decir, sin utilizar ningún algoritmo o procedimiento criptográfico.

Esta forma de recuperar la imagen original mediante la superposición de transparencias lleva aparejado el hecho de que la imagen recuperada tiene menos calidad y peor definición que la imagen secreta original. Un ejemplo de este tipo de criptografía puede verse en las figuras 19 y 20, donde se muestran, respectivamente, la imagen original junto con la recuperada y las dos sombras generadas en este caso.

FIGURA 19

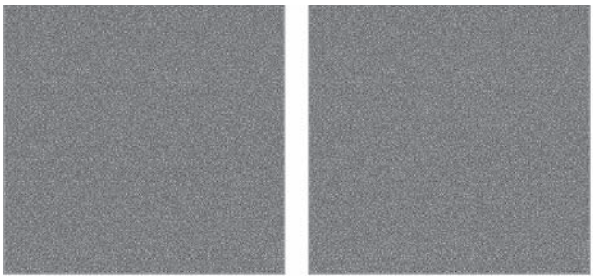
Imagen original e imagen recuperada al superponer dos transparencias.



FUENTE: ELABORACIÓN PROPIA A PARTIR DE UN CUADRO DE PICASSO.

FIGURA 20

Sombras de la imagen original impresas en sendas transparencias.



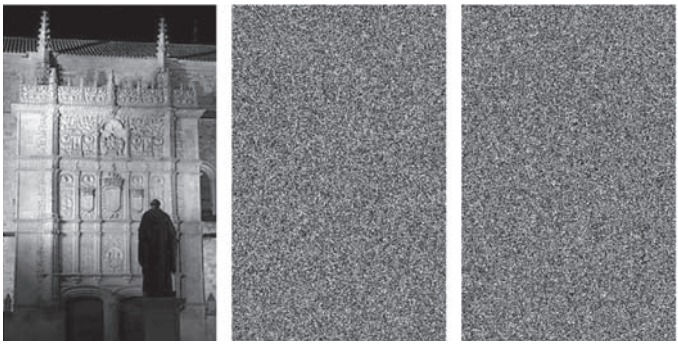
FUENTE: ELABORACIÓN PROPIA.

Es posible llevar a cabo el diseño de protocolos basados en la criptografía visual pero haciendo uso de otras herramientas matemáticas e implementarlos mediante programas

informáticos. Con ello se pueden evitar algunos de los problemas mencionados anteriormente, como la pérdida de calidad y resolución, haciendo que la imagen recuperada sea exactamente igual a la original. Un ejemplo de este tipo de protocolo puede verse en la figura 21.

FIGURA 21

Fachada de la Universidad de Salamanca y sombras en formato digital.



FUENTE: FOTOGRAFÍA REALIZADA E IMÁGENES ELABORADAS POR EL AUTOR.

Ataques a dispositivos físicos

Cada día aparecen dispositivos más pequeños que mantienen o superan las capacidades de cómputo de los que ya están en el mercado. Esto es especialmente significativo en el caso de los dispositivos de identificación por radio frecuencia (RFID o *Radio Frequency IDentification*), las tarjetas inteligentes (*smart card*, en inglés) y las tarjetas empleadas en la telefonía móvil, conocidas como módulo de identificación de abonado o tarjetas SIM (*Subscriber Identity Module*). Además, los chips incluidos en algunas de estas tarjetas son capaces de implementar operaciones criptográficas, ya sea para el cifrado de datos o para la firma electrónica.

Hasta hace unos años, la seguridad de los protocolos criptográficos se basaba, fundamentalmente, en la ofrecida

por los desarrollos teóricos que estaban justificados por la dificultad de resolver los problemas matemáticos subyacentes. Por esta razón, los ataques contra estos sistemas se dirigían a romper los algoritmos y problemas utilizados. Sin embargo, hoy se sabe que no es suficiente con que los algoritmos sean fuertes y seguros: es preciso que la implementación de los mismos sea también segura, de modo que no sea posible atacar los dispositivos debido a implementaciones erróneas o ineficientes (Fúster *et al.*, 2012, cap. 11).

Así pues, dado que la parte teórica puede suponerse segura, si se siguen las recomendaciones que pueden considerarse estándares, los nuevos métodos de ataque se dirigen hacia las implementaciones de las primitivas criptográficas en los dispositivos, por lo que tales ataques se conocen como ataques físicos. Entre ellos se distinguen dos tipos de ataques: los *ataques por canal lateral* y los ataques por *inducción de fallos*.

Estos ataques dependen del dispositivo y de la implementación en concreto, dándose por hecho que el atacante tiene acceso completo al dispositivo físico, conoce los algoritmos implementados, puede ejecutar dichos algoritmos cuantas veces necesite y es capaz de medir determinados parámetros generados por el dispositivo. Lo único que no conoce es la clave criptográfica almacenada en el dispositivo, que es, precisamente, su objetivo.

Los ataques por canal lateral se clasifican en función del tipo de parámetro que se mida a través del canal que se emplee para atacar el dispositivo. Así, se puede atacar un dispositivo, por ejemplo, por análisis temporal, por análisis de potencia o por análisis de emanaciones electromagnéticas. A continuación se presentan de forma muy esquemática algunos de estos posibles ataques.

Un ataque por *análisis temporal* es aquel que intenta obtener información sobre un algoritmo criptográfico, particularmente la clave almacenada, midiendo el tiempo que tarda en ejecutarse en el dispositivo en el que está implementado (Kocher, 1996), que está en función del número de operaciones que se ejecutan. Tal número depende, en general, del número de bits

igual a 1 que tiene la clave, es decir, el tiempo de computación suele ser mayor cuantos más bits iguales a 1 tenga la clave.

En los ataques por *análisis de potencia consumida* (Kocher *et al.*, 1998), se toma como premisa el hecho de que la potencia que consume un dispositivo, cuando está haciendo cálculos, depende no solo del número de operaciones que se ejecuten, sino también del tipo de operación en cuestión. Así, en general, hacer varias sumas consume más potencia que una única suma, o una suma consume menos potencia que una multiplicación.

En los *análisis simples de potencia* (SPA o *Simple Power Analysis*) se captura una o solo unas pocas trazas de consumo de potencia mientras el dispositivo criptográfico está funcionando. Para ello, basta con colocar una resistencia en serie con el dispositivo y capturar la traza con un osciloscopio.

La traza mostrará en qué momentos el consumo de potencia es más elevado que en otros.

Como hemos explicado, el proceso de descifrado con el sistema RSA consiste en realizar la operación $c^a \pmod{n}$, siendo a la clave privada. Dado que la forma más eficiente de calcular de la exponenciación modular anterior consiste en utilizar el algoritmo de elevar al cuadrado y multiplicar (véase "Algunos conceptos matemáticos" en el capítulo 3), tenemos todas las herramientas para analizar la traza obtenida.

En el algoritmo mencionado, si el bit del exponente (clave) que se procesa es un 0, se lleva a cabo un cuadrado, esto es, una multiplicación; pero si el bit procesado es un 1, además de la multiplicación correspondiente al cuadrado, se ejecuta una multiplicación adicional, por lo que el consumo de potencia es mayor en este segundo caso. Este sencillo análisis pone de manifiesto que una zona de bajo consumo de potencia seguida de otra de también bajo consumo equivale a un bit 0 en la clave, mientras que una zona de bajo consumo seguida de una de consumo alto representa un bit 1 de la clave. Analizando la traza completa se obtendrá entonces la expresión binaria de la clave privada, que era el objetivo del atacante.

Hay ocasiones en que debido a errores de medida, a la poca potencia de la señal recibida o a la presencia de ruido, por ejemplo, no se aprecia una clara relación entre la traza capturada y la potencia consumida. En estos casos, no es posible llevar a cabo un ataque como el anterior y se recurre a un *análisis diferencial de potencia* (DPA o *Differential Power Analysis*). Este ataque consiste en medir las trazas correspondientes a un gran número de cifrados o descifrados con diferentes valores como entrada y luego aplicar distintas técnicas estadísticas para intentar obtener la clave que se empleó.

En los ataques por *análisis de emanaciones electromagnéticas* (EMA o *ElectroMagnetic Analysis*) se utilizan este tipo de emanaciones que producen los circuitos de los chips debidas al desplazamiento de las cargas eléctricas en las pistas de las capas de metal, y que son medidas por sondas que se colocan cerca del chip (Quisquater *et al.*, 2001).

El análisis de la información conseguida se procesa de forma parecida a como se hace en los ataques SPA y DPA, dando lugar a dos tipos de ataques: *análisis electromagnético simple* (SEMA o *Simple ElectroMagnetic Analysis*) y *análisis electromagnético diferencial* (DEMA o *Differential ElectroMagnetic Analysis*).

En los *ataques por inducción de fallos* se intenta manipular el dispositivo que lleva a cabo las operaciones criptográficas con el fin de modificar su funcionamiento normal y obtener como salida un mensaje de error o un resultado erróneo (Bar-El *et al.*, 2006). En cualquier caso, comparando la salida obtenida al inducir un fallo y en una ejecución normal, es posible obtener información sobre la clave utilizada.

Este tipo de ataque suele ser más complicado de llevar a cabo que los anteriores, dado que requiere de un enorme trabajo de ingeniería inversa, de un equipamiento muy costoso y de un conocimiento muy preciso del procesador empleado y de sus componentes y celdas de memoria. La forma de inducir fallos puede ser alterando las condiciones de funcionamiento del chip, como hacerle funcionar fuera de

sus márgenes permitidos de temperatura, fuera de sus niveles aceptados de tensión, modificando la frecuencia de su reloj, disparando un láser en determinadas zonas de memoria para modificar la información almacenada o para alterar el flujo de un algoritmo, etc.

En algunas ocasiones es posible llevar a cabo un ataque combinando algunos de los ataques mencionados anteriormente, de modo que parte de la información proporcionada por uno de ellos pueda ser utilizada en otro.

La defensa contra los ataques anteriores, o la disminución de su efectividad, se lleva a cabo añadiendo *contramedidas* a los chips de las tarjetas, de modo que se evite, en la medida de lo posible, la pérdida de información.

Algunas de estas contramedidas consisten en introducir saltos o retrasos deliberados en los algoritmos de modo que el chip lleve a cabo operaciones que no son necesarias y tarde el mismo tiempo en ejecutar los cómputos, independientemente del número de ceros que tenga la clave. Otras tratan de evitar que el consumo de potencia dependa de los valores intermedios del algoritmo que se ejecuta y dificultar que el atacante pueda conseguir información útil. También hay contramedidas que intentan proteger el chip o determinadas partes del mismo, mediante rejillas o apantallamientos, de modo que se haga muy difícil acceder a sus partes más sensibles y se evite la inducción de fallos en su comportamiento.

Criptografía cuántica

Para concluir este capítulo, mencionaremos las últimas tendencias entre las que se mueve la criptografía a raíz de la posible llegada de la denominada *criptografía cuántica* (Brassard, 1988). Conviene distinguir entre esta y la *computación cuántica*. La segunda aborda y trata de resolver el problema de diseñar un ordenador capaz de hacer operaciones basadas en los principios de la física cuántica, mientras que la criptografía cuántica trataría de llevar a cabo los mismos servicios y

aplicaciones que la criptografía actual mediante el uso de la computación cuántica.

Existe una tendencia que da por hecho que la llegada de los computadores cuánticos acabará con la criptografía tal y como la conocemos actualmente, debido a la enorme capacidad de computación de estos y a la publicación, en 1997, de un artículo de Peter W. Shor (1959-) en el que se proponen sendos algoritmos capaces de factorizar números grandes y calcular logaritmos discretos en tiempo polinómico con la ayuda de un computador cuántico (Shor, 1997).

La tendencia que se opone a la anterior supone que los mismos computadores cuánticos podrían dan lugar a nuevos problemas que tales computadores no podrían resolver en un tiempo asequible, como sucede con los computadores actuales.

En cualquier caso, el posible desarrollo de la computación cuántica ha dado lugar al estudio de nuevos tipos de criptosistemas, cuya seguridad se basa en problemas matemáticos difíciles de resolver, pero no relacionados con los problemas de la factorización y del logaritmo discreto.

Con relación a la criptografía cuántica, hemos de indicar que hasta la fecha no se ha propuesto ningún sistema de cifrado cuántico, salvo que se considere como tal a la versión cuántica del cifrado de Vernam (véase "Cifrados en flujo" en el capítulo 4), es decir, utilizar una clave tan larga como el mensaje a cifrar y solo una vez, lo que requiere, además, un sistema seguro para la transmisión de la clave entre las dos partes que se intercambian la información.

Lo que sí se ha propuesto son protocolos para la distribución cuántica de claves o QKD (*Quantum Key Distribution*) basados en los principios de la física cuántica.

Comentaremos de forma muy resumida cómo se lleva a cabo la transmisión cuántica de claves. Ya se ha mencionado que la forma tradicional de representar la información es mediante bits, es decir, ceros y unos, lo que está relacionado con los dos estados posibles de un circuito eléctrico: abierto o cerrado. Por su parte, la información en la mecánica cuántica también se representa mediante valores discretos, como

sucede en la transmisión de fotones polarizados, ya sea a través de fibra óptica o del espacio libre.

Los principios de la física cuántica establecen que todo *cuanto* (cantidad mínima de materia, energía, etc.) está a la vez en varios estados superpuestos de forma indefinida hasta que es objeto de una medición (temperatura, polarización, etc.), momento en el que el resultado de la misma fija su estado, sin posibilidad de que el cuanto recupere ninguno de los estados que pudiera tener previos a la medición realizada (Teorema de imposibilidad de clonación).

La unidad mínima de información cuántica se conoce como *qubit* (de *quantum bit*, o bit cuántico) y puede estar, además de en los dos estados base tradicionales denotados por $|0\rangle$ y $|1\rangle$, en una superposición coherente de los dos, esto es, en el estado $|0\rangle$ y $|1\rangle$ simultáneamente. Se ha estimado que un computador cuántico, cuya longitud de palabra sea de 32 qubits, tendría una potencia de cálculo equivalente a la de unos $2^{32} \approx 4.300.000.000$ computadores actuales de, también, 32 bits.

Cuando una fuente luminosa en particular emite fotones, estos se encuentran polarizados de forma simultánea en todos los ángulos posibles, y al realizar una medición mediante un filtro polarizador, la polarización del fotón queda fijada con el mismo ángulo que tuviera el filtro. Si una vez fijada la polarización del fotón se lleva a cabo otra medición con un segundo filtro, el fotón atravesará este segundo filtro o no, dependiendo del ángulo que forme el segundo con respecto al primero. Dicho de otro modo: los fotones atravesarán el segundo filtro de modo probabilístico.

Así, si el segundo filtro es paralelo al primero, el fotón siempre atravesará el polarizador (probabilidad 1), pero si es perpendicular, el fotón no lo atravesará jamás (probabilidad 0). Entre estos dos extremos de 0º (paralelo) y 90º (perpendicular) caben todos los ángulos posibles, es decir, la probabilidad de atravesarlo estará entre 1 y 0, de modo que el resultado de la medición no será conocido hasta que esta no se lleve a cabo. Por ejemplo, si el ángulo formado por el primer

y el segundo filtro es de 45º, la probabilidad de que el fotón atraviese el segundo filtro es del 50%. Si es así, es decir, si el fotón atraviesa el segundo filtro, su polaridad quedará fijada según el ángulo de este segundo filtro.

Basado en estas propiedades de los fotones, el protocolo propuesto por Charles W. Bennet (1943-), conocido como B92 (Bennet, 1992), permite la distribución cuántica de una clave, eso sí, con el apoyo final de un canal de comunicación estándar.

El protocolo utiliza dos sistemas de referencia perpendiculares incompatibles, por ejemplo, un sistema con filtros en las posiciones vertical(↕)/horizontal(↔) y el otro sistema con filtros en las posiciones diagonal a +45º(↗)/diagonal a –45º(↖), y transmite fotones individuales polarizados. Los dos usuarios que desean utilizar este protocolo disponen de un canal cuántico por el que viajará la información mediante fotones.

Como paso previo al protocolo, Alicia genera de forma aleatoria una secuencia de bits y la almacena de modo secreto, que será de donde saldrá la clave a compartir con Bernardo.

En el protocolo, Alicia utiliza, por ejemplo, los filtros polarizados vertical (↕) y diagonalmente a +45º (↗), considerándolos como una forma de codificación para enviar el bit 0 y el bit 1, respectivamente, de la secuencia que generó inicialmente. Esta polarización y codificación debe ser conocida por Bernardo (no la secuencia de bits) con el fin de que este elija su sistema de referencia perpendicular al de Alicia. Así, Bernardo emplea los filtros polarizados horizontal (↔) y diagonalmente a –45º (↖). Además, Bernardo tiene que generar aleatoriamente una sucesión de estas polarizaciones para la posible recepción de los fotones que envíe Alicia. Una forma de llevarlo a cabo puede ser generar una secuencia de bits y codificar ambos filtros, de modo que el horizontal lo emplee para codificar un 1 y el segundo para codificar un 0. Además de este acuerdo, ambos deben establecer otros parámetros, como el número de fotones a enviar, etc.

Debido a la incertidumbre cuántica comentada anteriormente, Bernardo solo recibirá algunos de los fotones emitidos por Alicia. Así, Bernardo nunca recibirá un fotón si el bit de Alicia es un 0 (polarización vertical,↕) y el de Bernardo un 1 (polarización horizontal,↔) o si el de Alicia es un 1 (polarización diagonal a +45º, ↗) y el de Bernardo un 0 (polarización diagonal a –45º, ↖), debido a las posiciones perpendiculares de los polarizadores que utilizan cada uno de ellos. Cualquier otra posición de los filtros dará lugar, probabilísticamente, a la recepción de un fotón.

De este modo, si Bernardo no recibe, en una unidad de tiempo, ningún fotón deducirá que o bien su filtro estaba colocado perpendicularmente al considerado por Alicia (↕/↔ o ↗/↖) o bien formaba con este un ángulo de 45º (↕/↖ o ↗/↔), pero no llegó porque fue bloqueado por el filtro, dado que la probabilidad de no pasar es del 50%.

Por el contrario, si Bernardo recibe un fotón se debe a que sus filtros forman un ángulo de 45º con respecto a los de Alicia y el fotón no fue bloqueado por el filtro, de nuevo porque la probabilidad de pasar es del 50%. Así, si la polarización empleada por Bernardo es diagonal a –45º (↖), puede asegurar que Alicia empleaba un filtro polarizado verticalmente (↕), es decir, estaba considerando un bit 0. En el caso de que el filtro empleado por Bernardo estuviera polarizado horizontalmente (↔), Bernardo puede garantizar que el filtro de Alicia estaba polarizado diagonalmente a +45º (↗) y el bit de su secuencia en ese momento era un 1.

Cuando Bernardo recibe un fotón anota el hecho como "acierto" y toma nota del bit de su secuencia, que corresponde a la polarización que emplea en ese momento. Una vez transmitidos todos los fotones, Bernardo comunica a Alicia, por el canal convencional, las posiciones de su lista que corresponden a sus aciertos, pero no las polarizaciones utilizadas, con lo que Alicia y Bernardo compartirán como clave los bits de tales posiciones.

Bit de la secuencia de Alicia	0	1	0		1	
Polarización de Alicia	↕	↗	↕		↗	
Bit de la secuencia de Bernardo	1	0	0		1	
Polarización de Bernardo	↔	↖	↖		↔	
¿Llega un fotón a Bernardo?	No	No	No	Sí	No	Sí
Clave (bit) compartida				0		1

De todo lo anterior se puede deducir que la probabilidad de que a Bernardo le llegue un fotón es del 25% (la mitad de los cuales serán 0 y la otra mitad 1), por lo que la probabilidad de que al final del protocolo compartan la mitad de los bits transmitidos es del 25%.

Si un adversario, Carlos, pudiera escuchar la comunicación entre Bernardo y Alicia cuando el primero comunica las posiciones de sus aciertos, solo conocerá tales posiciones, pero no obtendrá información alguna sobre los bits.

El hecho de que Carlos pueda interceptar los fotones que Alicia envía a Bernardo no le ayuda a conocer la clave que compartirán Alicia y Bernardo. Veamos la razón con un ejemplo: supongamos que Alicia envía un fotón polarizado verticalmente (↕), de modo que está considerando un bit 0 de su secuencia, y Carlos, que tiene los mismos filtros que Bernardo, utiliza un filtro polarizado a –45° (↖) de modo que el fotón no le llega. En este caso, Carlos no tiene forma de saber cuál era el filtro de Alicia, si el polarizado a derecha +45° (↗), en cuyo caso nunca le hubiera llegado el fotón dado que esta probabilidad sería 0, o con polaridad ↕, pero de los que no llegan el 50%. Si Carlos hace uso de la máxima verosimilitud, supondrá que Alicia usó la polarización ↗ y enviará un fotón a Bernardo con esta polaridad.

Ahora bien, si Bernardo utiliza el polarizador ↖ no recibirá el fotón y Carlos no sacará provecho de ello, pero si usa el polarizador ↔ entonces recibirá un fotón con una probabilidad del 50%, pero, en este caso, Bernardo anotará un 1, precisamente el bit contrario al que envió Alicia, con lo que habrá una discrepancia entre los bits de Alicia y Bernardo.

Según hemos visto antes, o Bernardo no recibe un fotón o, si lo recibe, el bit que considera es exactamente el mismo que el de Alicia, pero no puede haber discrepancias. Usando este hecho, ambos pueden detectar la presencia de Carlos, tan solo con buscar posibles discrepancias entre los bits acordados. Para ello, eligen al azar algunas posiciones de la secuencia de bits acordada y se intercambian estos bits. Si la subsecuencia de estos bits no coincide exactamente, Alicia y Bernardo pueden asegurar que un adversario ha intervenido el envío de fotones y descartarlos todos. En otro caso, pueden repetir este proceso de intercambio varias veces para incrementar la probabilidad de detectar intrusiones o directamente pueden utilizar el resto de los bits intercambiados en el protocolo como clave.

A lo largo de este libro, hemos dado un repaso a las principales características y propiedades de la criptografía, justificando los métodos que han cubierto la necesidad que el hombre ha tenido desde siempre de mantener o comunicar información secreta. Así, se han presentado los sistemas de cifrado más importantes utilizados por el hombre a lo largo de la historia, desde la época clásica griega hasta nuestros días, pasando por las máquinas cifradoras más importantes inventadas para este fin, y presentando algunas de las líneas de desarrollo futuro de la criptografía.

Los métodos de cifrado actuales, ya sean simétricos o asimétricos, hacen uso de técnicas computacionales y herramientas matemáticas, por lo que ha sido imprescindible hacer referencia a ellas en determinados capítulos. En todo caso, las mismas han sido limitadas a los contenidos imprescindibles para un adecuado seguimiento del libro. Así, el lector podrá incluirlas en su bagaje cultural con el fin de conocer los medios que le permiten proteger su información privada y sus datos personales.

Glosario de algunos términos matemáticos utilizados en este libro

Aritmética modular: operaciones que se llevan a cabo en $\mathbf{Z}_n$, de modo que, una vez realizada la operación como números enteros, se divide entre n el valor obtenido y se considera como resultado el resto de la división efectuada.

Bit (digito binario o *binary digit*): dígitos utilizados en base 2 o binaria, es decir, 0 y 1.

Byte: conjunto de 8 bits.

Certificado electrónico (digital): fichero, emitido por una autoridad reconocida, por el que se garantiza que la clave pública y el resto de información contenida en el mismo pertenecen al usuario que se especifique.

Criptoanálisis: ciencia que trata de vulnerar los métodos de cifrado establecidos por la criptografía.

Criptografía: ciencia que trata de cómo intercambiar información de forma segura, haciendo el mensaje ilegible, sin ocultar la existencia de dicho mensaje.

Criptografía asimétrica (de clave pública): aquella que utiliza dos claves diferentes, una para cifrar (públicamente conocida) y otra para descifrar (secreta).

Criptografía simétrica (de clave secreta): aquella que utiliza la misma clave para cifrar y para descifrar y es compartida en secreto por emisor y receptor.

Criptología: ciencia resultante de la unión de la criptografía y el criptoanálisis.

Cuerpo: conjunto dotado con dos operaciones, suma y multiplicación, de modo que es un grupo aditivo conmutativo; el conjunto sin el cero es un grupo multiplicativo conmutativo y la multiplicación es distributiva respecto a la suma.

Curva elíptica: curva plana que viene definida por la ecuación de Weierstrass, cuya expresión es de la forma: $y^2 + a_1\, x\, y + a_3\, y = x^3 + a_2\, x^2 + a_4\, x + a_6$.

Elemento inverso a uno dado: elemento que al ser multiplicado por el primero da como resultado el neutro de la multiplicación, es decir, el 1. El inverso de a es a^{-1}.

Elemento neutro: elemento que operado con otro dado proporciona como resultado este último elemento. Con la operación de suma el neutro es el 0 y con la multiplicación es el 1.

Elemento opuesto a uno dado: elemento que al ser sumado con el primero da como resultado el neutro de la suma, esto es el 0. El opuesto de a es $-a$.

Esteganografía: ciencia que trata de la escritura de un mensaje, de modo que este quede encubierto u oculto.

Factorial de un número entero: producto de todos los números comprendidos entre el 1 y dicho número, es decir, $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$.

Firma electrónica (digital): protocolo criptográfico que permite demostrar la autoría y autenticidad de un documento digital. Corresponde a la versión electrónica de la firma manuscrita y tiene la misma validez legal es que esta.

Función resumen: función públicamente conocida que transforma una información de cualquier tamaño en una información con un tamaño prefijado de antemano.

Grupo: conjunto dotado de una operación que es interna, asociativa, tiene elemento neutro y cada elemento tiene su opuesto o inverso. El grupo es conmutativo si la operación lo es.

Grupo cíclico: grupo en el que existen elementos (llamados generadores) que al ser operados consigo mismos reiteradamente dan lugar a todo el grupo.

Indicador de Euler: función que proporciona la cantidad de números primos con un número dado y menores que tal número, $\phi(n)$.

Logaritmo de un número en base n, $\log_n(z) = x$: potencia a la que hay que elevar n para obtener z: $n^x = z$.

Máximo común divisor de dos números: mayor divisor que tienen en común dichos números: $\text{mcd}(a,b)$.

Número compuesto: todo número entero positivo no primo.

Número primo: todo número entero positivo cuyos únicos divisores son la unidad y él mismo.

Números primos entre sí (coprimos): números enteros positivos, primos o no, que no tienen divisores comunes salvo el 1, es decir, cuyo máximo común divisor es el 1.

Operación XOR (disyunción exclusiva o suma módulo 2): operación a nivel de bits que tiene la siguiente tabla de sumar: $0+0 = 0$, $0+1 = 1$, $1+0 = 1$, $1+1 = 0$.

Problema de Diffie-Hellman (DHP): calcular eficientemente el valor de $g^{a \cdot b}$ conocidos el primo p y los elementos g, g^a y g^b del grupo multiplicativo $\mathbf{Z}_p^*$, siendo g un generador del grupo y a y b dos números enteros positivos.

Problema de la factorización entera: determinar la descomposición en factores primos de un número compuesto.

Problema de la primalidad: decidir si un número grande dado es o no primo.

Problema del logaritmo discreto (DLP): calcular eficientemente el valor del número entero positivo a, conocidos el primo p y los elementos g y g^a del grupo multiplicativo $\mathbf{Z}_p^*$, siendo g un generador del grupo.

Qubit (bit cuántico o *quantum bit*): unidad mínima de información cuántica que puede estar, además de en los dos estados base tradicionales denotados por $|0\rangle$ y $|1\rangle$, en una superposición coherente de los dos, esto es, en el estado $|0\rangle$ y $|1\rangle$ simultáneamente.

Símbolos

$\mathbf{Z}$: conjunto de los números enteros.

$\mathbf{Z}_n$: conjunto de los posibles restos que se obtienen al dividir un entero entre n, es decir, $\mathbf{Z}_n = \{0, 1, 2\ldots, n-1$.

$\mathbf{Z}_n^*$: conjunto de los números enteros que tienen inverso en $\mathbf{Z}_n$.

$a \equiv b \pmod{n}$ relación entre los enteros a y b que tienen la propiedad de poseer el mismo resto al ser divididos por n, o lo que es igual, que su diferencia es un múltiplo de n.

$a^k \pmod{n}$ representa la exponenciación modular, es decir, la operación de multiplicar a por sí misma k veces en $\mathbf{Z}_n$, esto es, elevar a a la k y hacer módulo n.

Bibliografía

Agrawal, M.; Kayal, N. y Saxena, N. (2004): "Primes is in P", *Annals of Mathematics,* 160, 2, pp. 781-793.

Bar-El, H.; Choukri, H.; Naccache, D.; Tunstall, M. y Whelan, C. (2006): "The sorcerer's apprentice guide to fault attacks", *Proc. of IEEE*, 94, 2, pp. 370-382.

Bennett, C. H. (1992): "Quantum cryptography using any two nonorthogonal states", *Phys. Rev. Lett.*, 68, pp. 3121-3124.

Brassard, G. (1988): *Modern Cryptology: a tutorial*, Lecture Notes in Computer Science, 325, Springer-Verlag, Berlín.

Cortes Generales (2003): "Ley de firma electrónica", Ley 59/2003, *BOE* número 304 de 20 de diciembre de 2003.

Crespo Sánchez, J.; Espinosa García, J.; Hernández Encinas, L.; Rifá Pous, H. y Torres Hernández, M. (2006): "Hacia una nueva identificación electrónica del ciudadano: el DNI-e", Actas de la IX Reunión Española de Criptología y Seguridad de la Información (IX RECSI), pp. 660-673.

De León, M. y Timón, A. (2014): *Rompiendo códigos. Vida y legado de Alan Turing*, CSIC-Los Libros de la Catarata, Madrid, p. 48.

De Rojas, F. (2013): *La Celestina*, Alianza Editorial, Madrid.

Della Porta, G. B. (1996): *De ocultis literarum notis-vulgo de ziferis*, facsímil de la edición de 1593, Universidad de Zaragoza, Zaragoza [se reproduce la siguiente edición: *De occultis*

literarum notis seu artis animi sensa occulte aliis significandi, aut ab aliis significata expiscandi enodandique libri IIII, Montisbeligardi (Montbéliard), Francia, 1593].

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

Doyle, A. C. (2013): *El misterio de los bailarines*, Vicens-Vices, Madrid.

Durán Díaz, R. *et al.* (2005): *El criptosistema RSA*, RA-MA, Madrid.

ElGamal, T. (1985): "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transaction on Information Theory,* 31, pp. 469-472.

Fúster Sabater, A.; De la Guía Martínez, D.; Hernández Encinas, L.; Montoya Vitini, F. y Muñoz Masqué, J. (2004): *Técnicas criptográficas de protección de datos*, 3ª ed, RA-MA, Madrid.

Fúster Sabater, A.; Hernández Encinas, L.; Martín Muñoz, A.; Montoya Vitini, F. y Muñoz Masqué, J. (2012): *Criptografía, protección de datos y aplicaciones. Una guía para estudiantes y profesionales,* RA-MA, Madrid.

Golomb, S. W. (1982): *Shift Register Sequences*, Aegean Park Press, Revised ed., Laguna Hills, CL.

Heródoto (1985): *Historia*, Gredos, Madrid.

Kocher, P. C. (1996): "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", *Lecture Notes Comput. Sci.*, 1109, pp. 104-113.

Kocher, P.; Jaffe, J. y Jun, B. (1998): "Introduction to differential power analysis and related attacks", *Technical report*, Cryptography Research Inc.

Menezes, A.; Van Oorschot, P. y Vanstone, S. (1997): *Handbook of applied cryptography*, CRC Press, Boca Raton, FL.

Mowry, D. P. (2014): "German ciphers machines of World War II", Center for Cryptologic History, National Security Agency, USA, Revised edition.

Plinio el Viejo (2007): *Historia Natural*, Cátedra, Madrid.

Poe, E. A. (2007): *El escarabajo de oro*, Nivola, Madrid.

Polibio (1997): *Historias*, Gredos, Madrid.

Quisquater, J.-J. y Samyde, D. (2001): "ElectroMagnetic Analysis (EMA): Measures and counter-measures for smart cards", *Lecture Notes Comput. Sci.* 2140, pp. 200-210.

RAE (2015): *Diccionario de la lengua Española*, 23 edición (disponible en http://www.rae.es).

Rivest, R.; Shamir, A. y Adleman, L. M. (1978): "A method for obtaining digital signatures and public-key cryptosystems", *Communications ACM 21*, 2, pp. 120-126.

Sgarro, A. (1990): *Códigos secretos*, Pirámide, Madrid.

Shor, P. W. (1997): "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". *SIAM J. Computing*, 26, 5, pp. 1484-1509.

Singh, S. (2000): *Los códigos secretos*, Debate, Madrid.

Soler Fuensanta, J. R.; López-Brea Espiau, F. J. y Weierud, F. (2010): "Spanish Enigma: A History of the Enigma in Spain", *Cryptologia*, 34, 4, pp. 301-328.

Turing, A. (1936): "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society* 42, 2, pp. 230-265.

Van Tilborg, H.C.A. (ed.) (2005): *Encyclopedia of Cryptography and Security*, Springer, Nueva York.

Verne, J. (1987): *Matías Sandorf*, Porrúa, México.

— (2004): *Viaje al centro de la Tierra*, Anaya, Madrid.

Weierud, F. (2000): "Sturgeon, The FISH BP Never Really Caught", *Coding Theory and Cryptography,* pp. 18-52, Springer, Berlín.

Páginas web

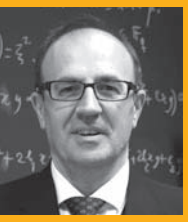
http://ciphermachines.com/
http://cryptologicfoundation.org/
http://enigmaco.de/enigma
http://users.telenet.be/d.rijmenants/en/enigmasim.htm
http://users.telenet.be/d.rijmenants/en/timeline.htm
http://www.bbc.co.uk/history/topics/enigma
http://www.codesandciphers.org.uk/
http://www.codesandciphers.org.uk/enigma/

http://www.criptored.upm.es/software/sw_m006a.htm
http://www.cryptomuseum.com/
http://www.dnielectronico.es/
http://www.dnielectronico.es/servicios_disponibles/index.html
http://www.kriptopolis.com/enigma
https://www.cryptool.org/
https://www.nsa.gov/about/cryptologic_heritage/museum/

¿QUÉ SABEMOS DE?

La criptografía

Desde siempre, el hombre ha sentido la necesidad de tener secretos y guardarlos a buen recaudo. Tan solo en algunas situaciones ha deseado compartirlos con determinados amigos o aliados, asegurándose de que aquellos no eran conocidos por terceras partes. Una de las formas que ideó fue la transformación del contenido de mensajes siguiendo determinadas reglas que modificaban la información del mensaje, de modo que, aplicando las reglas inversas o adecuadas, sería posible recuperar el mensaje original. El objetivo de la obra es dar a conocer algunas de las herramientas más utilizadas en la sociedad de la información para lograr la confidencialidad, integridad y autenticidad de la información mediante los métodos de cifrado de la criptografía. Los temas se abordan paralelamente al desarrollo de la historia de esta ciencia, comenzando con la época clásica griega, pasando por la Segunda Guerra Mundial, hasta llegar a la criptografía empleada hoy en día.



Luis Hernández Encinas es licenciado y doctor en Ciencias Matemáticas por la Universidad de Salamanca. Es investigador y actualmente director del Instituto de Tecnologías Físicas y de la Información “Leonardo Torres Quevedo” (ITEFI).

ISBN: 978-84-00-10045-2

GOBIERNO DE ESPAÑA · MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD · CSIC