Vol.8 / No.1 / Junio 2024, pp. 45-82
SahuarUS

Criptografía de los cifrados de bloque

Eduardo Velasco-Barreras
Departamento de Matemáticas, Universidad de Sonora.
eduardo.velasco@unison.mx

Resumen

En este trabajo discutiremos la importancia y el funcionamiento de los cifrados de bloque en criptografía. Particularmente, presentaremos algunas herramientas matemáticas en las que dichos cifrados están basados, y cómo la complejidad de las mismas ha logrado asegurar el nivel de seguridad que exigían las aplicaciones de su época. Entre los cifrados de bloque que se discutirán, se encuentra el *Data Ecryption Standard*, el cual fue el cifrado de bloque más utilizado desde la década de los ochenta hasta finales de los noventa.

Palabras Clave: Criptografía; Cifrados de bloque; Data Encryption Standard; Aritmética modular

DOI: 10.36788/sah.v8i1.100
Recibido: 23 de abril de 2019
Aceptado: 29 de junio de 2024

1. Motivación

La criptografía es la ciencia de la escritura secreta. Su objetivo es ocultar el significado de un mensaje, de manera que sólo la persona a quien dicho mensaje es enviado sea capaz de leerlo. Actualmente, la criptografía tiene muchísimas aplicaciones relacionadas con la vida cotidiana, como son el acceso seguro a páginas web, cifrado de correos electrónicos, sistemas de respaldo de archivos, firmas digitales, trasacciones finacieras, entre muchas otras.

A pesar de que el desarrollo que en las últimas décadas ha tenido la criptografía está relacionado con los saltos agigantados de la revolución tecnológica, su historia se remonta incluso hacia las civilizaciones más antiguas. En efecto, prácticamente cada civilización que ha desarrollado una forma de comunicación escrita ha utilizado también formas secretas de comunicación. Por ejemplo, en el antiguo pueblo egipcio de Menat Jufu, se han encontrado *jeroglíficos no estándares* escritos en la tumba del nomarca de Orix llamado Jnumhotep II. Dicho esquema de cifrado es lo que hoy en día llamaríamos *cifrado por sustitución*.

Otro ejemplo muy conocido de criptografía es el de la escítala, la cual era utilizada por los éforos (magistrados) de la ciudad de Esparta. La escítala consistía de una vara cilíndrica alrededor de la cual se enrollaba una tira de pergamino, de manera que al escribir sobre ella y posteriormente desenrollarla, se obtenía un mensaje ininteligible para un tercero que intentara leerlo. Para descifrar dicho mensaje, era necesario enrollar la tira de pergamino

ISSN 2448 5365
Universidad de Sonora

nuevamente en algún cilindro del mismo diámetro. Desde tiempos del historiador y filósofo griego Plutarco se creía que el propósito de la escítala era impedir a terceros conocer el contenido del mensaje. De esta manera, hablaríamos de un *cifrado por trasposición*, pues cada caracter (o bloques de caracteres) son desplazados siguiendo un patrón bien definido. Sin embargo, la revisión de autores más tempranos ha permitido concluir que la escítala no era utilizda como método de cifrado [8], sino que quizás era un *método de autenticación*, es decir, que su fin era validar la identidad del emisor del mensaje [13].

Figura 1: A la izquierda, una escítala en forma de cilindro hexagonal (creada por Eivind Lindbråten `https://commons.wikimedia.org/wiki/File:Skytale.png`). A la derecha, una forma simple de hacerla (`http://unmuseum.mus.pa.us/excoded.htm`).

Para cerrar esta parte introductoria, presentaremos un cifrado clásico que también es bastante conocido, pero que además será útil para los propósitos de esta exposición. Se dice que el político y militar romano Julio César se comunicaba con sus tropas usando un corrimiento de todas las letras del alfabeto por medio de un número fijo de pasos. Supongamos que, en lugar de utilizar como alfabeto las letras A, B, C, D, ..., W, X, Y, Z en el orden usual, tomamos el alfabeto empezando en alguna otra letra, digamos, l, m, n, ñ, ... , h, i, j, k. Es decir, que al escribir un mensaje, en lugar de utilizar la letra "A", usaríamos la letra "l"; en vez de escribir la letra "B", pondríamos "m" y así sucesivamente. En particular, al llegar al final del alfabeto se continúa desde el principio (la "O" se reemplaza por la "z" y la siguiente letra que es la "P" se reemplaza por la "a"). La tabla 1 muestra la clave completa de este ejemplo. Si el mensaje que queremos cifrar es "este texto es indescifrable", entonces, de acuerdo a dicha tabla, escribiríamos "ODEO EOIEZ OD SXÑODNSPCLMVO".

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o	p	q	r	s	t	u	v	w	x	y	z
L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K

Tabla 1: La clave completa de nuestro ejemplo de cifrado de Julio César.

Como un ejercicio para el lector, ¿pudiera descifrar el siguiente mensaje, sabiendo que fue cifrado con la clave anterior? Disculpen la falta de ortografía, pero nuestra clave no incluye caracteres con acento.

> AZÑOWZD LPSCWLC NLEOQZCSNLWOXEO BFO OV SXQOXSZ RFWLXZ XZ OD NLALK ÑO SXGOXELC XSXQFX NZÑSQZ DONCOEZ BFO OV ACZASZ SXQOXSZ RFWLXZ XZ AFOÑL ÑODNSPCLC. OÑQLC LVVLX AZO.

2. Aritmética modular

De los ejemplos presentados en la introducción pudiera tenerse la impresión de que las matemáticas no juegan un rol importante en la criptografía, o al menos no es muy transparente el cómo las matemáticas pueden ayudar a describir los cifrados presentados arriba o bien, generar otros más complejos. Resulta ser que la teoría de números, así como estructuras algebraicas como los grupos, anillos y campos, son esenciales en muchos de los cifrados que se usan hoy día.

Una de las herramientas fundamentales para la criptografía es la aritmética modular. Para el lector que no esté familiarizado con este tema, daremos aquí un pequeño repaso. Sin embargo, es recomendable que el lector pueda profundizar en un tema tan bonito, para lo cual sugerimos el libro [11, Capítulo 2]. En cambio, si el lector considera estar suficientemente familiarizado con las nociones básicas de aritmética modular y la noción de inversos multiplicativos, puede omitir el resto de esta sección y continuar la lectura en la sección 3.

2.1. Suma y multiplicación

Para un primer ejemplo de cómo funciona la aritmética modular, fijemos un número, por ejemplo, 6. Ahora supongamos que vamos a ordenar todos los números enteros en seis grupos, llamados *clases*, usando lo que se conoce por *relaciones de equivalencia*: Decimos que los enteros a y b pertenecen a la misma clase si $b - a$ es múltiplo de 6. Por ejemplo, 17 y 53 pertenecen a la misma clase, pues su diferencia es $53 - 17 = 36$, que es múltiplo de 6. Asimismo, 17 y -1 también pertenecen a la misma clase, pues $-1 - 17 = -18$, que es múltiplo de 6. No es difícil observar que todos estos números, -1, 17 y 53, pertenecen todos a la misma clase que el 5: las respectivas diferencias con 5 son -6, 12 y 48, los cuales son todos múltiplos de 6.

Una manera un poco más eficiente de proceder sería respondiendo a lo siguiente: ¿cuáles son todos los números enteros que pertenecen a la misma clase que el 5? Puesto que, por definición, hemos definido que todos los números de una misma clase tengan diferencia múltiplo de 6, podemos ir contando de 6 en 6 a partir de 5. De esta manera, recorreríamos hacia adelante los números 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, ... Similarmente, si nos vamos de 6 en 6 a partir de 5 pero hacia atrás, recorreríamos los números 5, -1, -7, -13, -19, ... En la tabla 2, los números de la clase de equivalencia del 5 se han colocado en el renglón superior. Asimismo, si empezamos desde el 0 y contamos de 6 en 6 hacia delante y hacia atrás, recorreremos los números que pertenecen al reglón inferior. De manera similar, podemos partir de los números 1, 2, 3 o 4 y contar de 6 en 6 hacia adelante y hacia atrás, obteniendo cada uno de los otros renglones de la tabla 2.

Ahora observemos la siguiente propiedad: si tomamos el 52 y hacemos su división entre 6, el resultado no es entero, sino que deja residuo 4, $52 = 6 \times 8 + 4$. Esto nos dice que el 52 pertenece a la misma clase que el 4, y que para llegar del 4 al 52 es hay que avanzar 8 lugares a la derecha. Similarmente, como 31 deja residuo 1 al dividirse entre 6, se tiene que 31 y 1 pertenecen a la misma clase.

Por otra parte, este acomodo tan sencillo de los números en renglones que van de 6 en

...	-19	-13	-7	-1	5	11	17	23	29	35	41	47	53	59	65	...
...	-20	-14	-8	-2	4	10	16	22	28	34	40	46	52	58	64	...
...	-21	-15	-9	-3	3	9	15	21	27	33	39	45	51	57	63	...
...	-22	-16	-10	-4	2	8	14	20	26	32	38	44	50	56	62	...
...	-23	-17	-11	-5	1	7	13	19	25	31	37	43	49	55	61	...
...	-24	-18	-12	-6	0	6	12	18	24	30	36	42	48	54	60	...

Tabla 2: Cada uno de los seis renglones es una clase de equivalencia módulo 6.

6 tiene varias cualidades interesantes. Tomemos un número del renglón verde, por ejemplo, el 44, y tomemos un número que pertenezca al reglón rojo, por ejemplo, el -13. Si sumamos ambos números obtenemos $44 + (-13) = 31$, que pertenece al renglón azul. Resulta que si tomamos cualquier otro par de números que también pertenezcan a los renglones verde y rojo, su suma siempre va a caer en el renglón azul (por ejemplo, 2+5=7). Esta propiedad no es exclusiva de los renglones verde y rojo. Si elegimos cualquier otro par de renglones, por ejemplo, azul y amarillo, las sumas entre elementos de dichos renglones siempre caerán en el renglón naranja.

Una manera de sintetizar lo anterior es la siguiente: denotemos por **0** a la clase del cero (al conjunto de todos los elementos del renglón blanco, o sea los múltiplos de 6). Asimismo, denotemos por **1** a la clase del 1 (el conjunto de todos los elementos del renglón azul). Similarmente, **2**, **3**, **4** y **5** denotarán a los renglones verde, amarillo, naranja y rojo. Puesto que la suma de cualquier elemento de la clase del 2 (renglón verde) con cualquier elemento de la clase del 5 (renglón rojo) pertenece a la clase del 1 (renglón azul), escribimos $\mathbf{2}+\mathbf{5}=\mathbf{1}$. Por otra parte, también tenemos que las sumas de elementos de la clase del 1 con elementos de la clase del 3 (azul y amarillo) da elementos de la clase del 4 (naranja), podemos escribir $\mathbf{1}+\mathbf{3}=\mathbf{4}$. Todas las posibles relaciones de sumas entre las clases **1**, **2**, **3**, **4**, **5** y **6** se resumen en la tabla de la suma que se presenta a continuación. Más aún, con la multiplicación también se tiene una operación bien definida, como se exhibe en la segunda tabla.

$+$	**0**	**1**	**2**	**3**	**4**	**5**
0	**0**	**1**	**2**	**3**	**4**	**5**
1	**1**	**2**	**3**	**4**	**5**	**0**
2	**2**	**3**	**4**	**5**	**0**	**1**
3	**3**	**4**	**5**	**0**	**1**	**2**
4	**4**	**5**	**0**	**1**	**2**	**3**
5	**5**	**0**	**1**	**2**	**3**	**4**

$\times$	**0**	**1**	**2**	**3**	**4**	**5**
0	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**1**	**2**	**3**	**4**	**5**
2	**0**	**2**	**4**	**0**	**2**	**4**
3	**0**	**3**	**0**	**3**	**0**	**3**
4	**0**	**4**	**2**	**0**	**4**	**2**
5	**0**	**5**	**4**	**3**	**2**	**1**

Estas tablas dan lugar a la *aritmética módulo 6*. Sin embargo, esto puede hacerse con cualquier otro número. Un ejemplo cotidiano lo tenemos en las horas del día, así como en los días de la semana.

Ejemplo 2.1 *Un joven emocionado por su próximo cumpleaños le dice a sus compañeros: "Faltan 531 horas para mi cumpleaños". Si esto ocurrió a las 7 de la tarde, ¿a qué hora nació el joven?*

Solución. En este ejemplo, aplicaremos aritmética módulo 24, ya que es el número de horas que tiene un día. A las 7 de la tarde son las **19** horas. Si agregamos **531** horas, tenemos que $\mathbf{19 + 531 = 540 = 12}$, pues el residuo que deja 540 al dividirse entre 24 es 12. Por lo tanto, el joven nació a las 12 del día. ■

Ejemplo 2.2 *Supongamos que un turista decide visitar nueve ciudades el mismo número de días cada una. Se sabe que a la primera ciudad llegó un viernes y se fue un miércoles, y que en el mismo día que se iba de una ciudad llegaba a la otra. ¿Qué día se fue el turista de la última ciudad?*

Solución. En este ejemplo carecemos de suficiente información para conocer cuántos días estuvo exactamente el turista en cada ciudad. Sin embargo, sí podemos dar respuesta a la pregunta anterior utilizando *aritmética módulo 7*. Como el turista llegó en viernes a la primera ciudad y se fue en miércoles, el turista pudo haber estado 5 días si no permaneció alguna semana completa, pero también pudo haber estado 12 días si se estuvo una semana completa, o bien 19 días si estuvo dos semanas completas, etc. En general, en número de días que el turista permaneció en la primera ciudad fue $7k + 5$, donde k es el número de semanas completas que el turista permaneció en dicho lugar. En otras palabras, el número de días que el turista permaneció en la primera ciudad es algún elemento de **5**, la clase del 5 en módulo 7. Dado que cada ciudad permaneció el mismo número de días, en total estuvo $\mathbf{5{\times}9{=}45{=}3}$ días sin contar las semanas. A partir del viernes, contando 3 días llegamos al lunes. Puesto que el viaje inició en viernes, el turista tuvo que irse un lunes de la última ciudad. ■

Estos dos ejemplos ilustran la manera intuitiva en que podemos pensar en la aritmética modular. Así como las horas del día y los días de la semana se repiten cíclicamente, también las operaciones en aritmética modular se repiten en ciclo.

Ejemplo 2.3 *Otro ejemplo cotidiano de aritmética modular es cuando trabajamos con números pares e impares. Sabemos que la suma de dos números pares da como resultado un número par, al igual que la suma de dos números impares. En cambio, cuando sumamos dos números de distinta paridad el resultado es impar. En cuanto a la multiplicación, el producto de dos números impares es impar, mientras que el producto con un número par siempre es par. Estos hechos pueden resumirse en las tablas de suma y multiplicación módulo 2 que presentamos a continuación. La clase* **0** *en módulo 2 consiste de todos los números pares, mientras que* **1** *consiste de todos los números impares.*

$+$	**0**	**1**
0	**0**	**1**
1	**1**	**0**

$\times$	**0**	**1**
0	**0**	**0**
1	**0**	**1**

La suma módulo 2 puede extenderse de manera natural a bloques de igual longitud. Por ejemplo, si $A = 1111010010101$ *y* $B = 0001101001110$*, entonces*

$$A \oplus B := 1111010010101 \oplus 0001101001110 = 1110111011011.$$

Esta suma, llamada suma exclusiva *o XOR, lo que hace es sumar módulo 2 cada una de las componentes.*

En general, dado un entero $n \in \mathbb{Z}$, consideremos el conjunto $n\mathbb{Z}$ de los enteros múltiplos de n. Luego, podemos dividir al anillo de los enteros $\mathbb{Z}$ en n clases de equivalencia dadas por la siguiente relación: $a \sim b$ si $b - a \in n\mathbb{Z}$. Las n clases de equivalencia son $\mathbf{0}, \mathbf{1}, \ldots, \boldsymbol{n-1}$ y denotaremos al conjunto de dichas clases de equivalencia por $\mathbb{Z}/n\mathbb{Z}$. Asimismo, usaremos la notación $a \mod n$ para referirnos al número entero entre 0 y $n-1$ que pertenece a la misma clase de equivalencia que a. Ahora bien, el hecho de que las operaciones de suma y multiplicación de clases de equivalencia estén bien definidas recae fundamentalmente en que el conjunto $n\mathbb{Z}$ es un *ideal* en el anillo $\mathbb{Z}$. Esto quiere decir que:

1. La suma de dos múltiplos de n es nuevamente un múltiplo de n: $n\mathbb{Z} + n\mathbb{Z} \subseteq n\mathbb{Z}$.
2. Al multiplicar un múltiplo de n por *cualquier otro entero* el resultado es múltiplo de n: $n\mathbb{Z} \times \mathbb{Z} \subseteq n\mathbb{Z}$.

En estos términos, el conjunto de clases de equivalencia $\mathbb{Z}/n\mathbb{Z}$ adquiere estructura de anillo con la suma y multiplicación de clases de equivalencia. La teoría de ideales dentro de las estructuras algebraicas son de gran importancia, no sólo en aritmética modular, sino para la construcción de otros ejemplos que se usan en criptografía. Por ello, sugerimos al lector interesado el profundizar en esta teoría, por ejemplo, en la referencia [6].

2.2. Inversos módulo n y el algoritmo euclidiano extendido

Consideremos el conjunto $\mathbb{Z}/n\mathbb{Z}$ de los enteros módulo n, y sea $\boldsymbol{a} \in \mathbb{Z}/n\mathbb{Z}$ la clase de equivalencia de algún entero $a \in \mathbb{Z}$. Una pregunta natural es cúando existe un entero $b \in \mathbb{Z}$ tal que $\boldsymbol{b} \in \mathbb{Z}/n\mathbb{Z}$ sea el inverso multiplicativo de $\boldsymbol{a}$. Por *inverso multiplicativo* queremos decir que la multiplicación de ambos sea $\mathbf{1}$, $\boldsymbol{ab} = \mathbf{1}$. En el lenguaje de anillos, se dice que $\boldsymbol{a}$ es una *unidad* en el anillo $\mathbb{Z}/n\mathbb{Z}$.

Por ejemplo, observemos la tabla de multiplicación módulo 6 que presentamos en la subsección anterior. Podemos ver que, como resultado de una multiplicación, el $\mathbf{1}$ solamente se obtiene de $\mathbf{1} \times \mathbf{1} = \mathbf{1}$ y $\mathbf{5} \times \mathbf{5} = \mathbf{1}$. En ese sentido, solamente $\mathbf{1}$ y $\mathbf{5}$ son unidades en $\mathbb{Z}/6\mathbb{Z}$. Observemos ahora la tabla de multiplicación en módulo 7. En este caso, podemos ver que en cada renglón, exceptuando el primero, aparece un $\mathbf{1}$, lo que nos dice que *todos los elementos distintos de cero de* $\mathbb{Z}/7\mathbb{Z}$ *son unidades*. Más precisamente, como $\mathbf{1} \times \mathbf{1} = \mathbf{1}$, $\mathbf{2} \times \mathbf{4} = \mathbf{1}$, $\mathbf{3} \times \mathbf{5} = \mathbf{1}$, $\mathbf{4} \times \mathbf{2} = \mathbf{1}$, $\mathbf{5} \times \mathbf{3} = \mathbf{1}$ y $\mathbf{6} \times \mathbf{6} = \mathbf{1}$, tenemos que $\mathbf{1}$ es su propio inverso, $\mathbf{2}$ y $\mathbf{4}$ son inversos uno del otro, $\mathbf{3}$ y $\mathbf{5}$ son inversos uno del otro y $\mathbf{6}$ es su propio inverso.

El hecho de que alguna clase residual $\boldsymbol{a}$ admita un inverso $\boldsymbol{b}$ en $\mathbb{Z}/n\mathbb{Z}$ depende tanto de $\boldsymbol{a}$ como de n. En términos de representantes de clase, es decir, trabajando con números enteros $a \in \boldsymbol{a}$ y $b \in \boldsymbol{b}$ en lugar de con sus clases de equivalencia, el que $\boldsymbol{a} \times \boldsymbol{b} = \mathbf{1}$ significa que $ab - 1$ sea múltiplo de n, es decir, $ab - 1 = nk$ para algún k. Esto último equivale a $ab - nk = 1$, es decir, a expresar al 1 como *combinación lineal* de a y n. Se puede probar que esto es posible siempre y cuando a y n sean *primos relativos* [11], es decir, que no tengan divisores comunes aparte de ± 1. Más precisamente, que el máximo común divisor de a y n sea 1, $(a, n) = 1$.

$\times$	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	**0**	**1**	**2**	**3**	**4**	**5**	**6**
2	**0**	**2**	**4**	**6**	**1**	**3**	**5**
3	**0**	**3**	**6**	**2**	**5**	**1**	**4**
4	**0**	**4**	**1**	**5**	**2**	**6**	**3**
5	**0**	**5**	**3**	**1**	**6**	**4**	**2**
6	**0**	**6**	**5**	**4**	**3**	**2**	**1**

Tabla 3: Tabla de multiplicación módulo 7.

Proposición 2.1 *Un entero a admite un inverso multiplicativo en módulo n si y sólo si a y n son primos relativos. Dicho de otra manera, la clase $\boldsymbol{a} \in \mathbb{Z}/n\mathbb{Z}$ es una unidad si y sólo si $(a, n) = 1$ para algún representante $a \in \boldsymbol{a}$.*

Utilizando el criterio anterior, podemos explicar por qué en la multiplicación módulo 6 solamente 1 y 5 tienen inverso: resulta que 1 y 5 no tienen factores en común con 6, mientras que 0, 2, 3, y 4 sí lo tienen. En el caso de la multiplicación módulo 7, todos los elementos distintos de cero tienen inverso porque 7 es número primo, y por lo tanto, los únicos enteros que tienen factores en común con 7 son los múltiplos de 7 (la clase **0**).

El algoritmo euclidiano. De acuerdo a la discusión anterior, la construcción de una clase $\boldsymbol{b}$ tal que $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{1}$ equivale a encontrar un entero b tal que $ab - nk = 1$ para alguna k, es decir, a expresar 1 como combinación lineal de a y n. Este procedimiento puede hacerse mediante el *algoritmo euclidiano extendido*.

Fijemos dos enteros n y a y supongamos $n > a > 0$. Recordemos que el *algoritmo de la división* nos dice que existen únicos enteros q y r tales que

$$n = aq + r, \qquad 0 \leq r < a.$$

Resulta que el máximo común divisor (n, a) coincide con el de (a, r), con la ventaja de que ahora los enteros involucrados son más pequeños: $n > a > r$. De esta manera, podemos repetir el algoritmo de la división para generar enteros cada vez más pequeños cuyo máximo común divisor sea siempre el mismo:

$$\begin{aligned}
n &= aq + r, & (a, r) &= (n, a);\\
a &= rq_1 + r_1, & (r, r_1) &= (a, r);\\
r &= r_1q_2 + r_2, & (r_1, r_2) &= (r, r_1);\\
r_1 &= r_2q_3 + r_3, & (r_2, r_3) &= (r_1, r_2);\\
r_2 &= r_3q_4 + r_4, & (r_3, r_4) &= (r_2, r_3);\\
&\vdots & &\vdots
\end{aligned}$$

Recordemos que en cada etapa hemos generado un residuo que es más pequeño tal que el máximo común divisor entre cualesquiera dos consecutivos es el mismo, $n > a > r > r_1 >$

$r_2 > \ldots$, $(r_i, r_{i+1}) = (n, a)$. Puesto que en cada etapa obtenemos un residuo más pequeño, eventualmente debemos obtener residuo cero:

$$r_{m-2} = r_{m-1}q_m + r_m,$$
$$r_{m-1} = r_m q_{m+1},$$

es decir, r_m es el último residuo no cero, y el siguiente residuo ya es cero, $r_{m+1} = 0$. Como

$$(n, a) = (r_m, r_{m+1}) = (r_m, 0) = r_m,$$

concluimos que el último residuo no cero, es decir r_m, es el máximo común divisor.

Proposición 2.2 (Algoritmo euclidiano) *Sean a y n enteros positivos. Al aplicar sucesivamente el algoritmo euclidiano, el último residuo no cero es el máximo común divisor (a, n).*

Ejemplo 2.4 *Consideremos $n = 2019$ y $a = 1115$. Aplicando el algoritmo euclidiano, obtenemos*

$$\begin{aligned}
2019 &= 1115 \times 1 + 904,\\
1115 &= 904 \times 1 + 211,\\
904 &= 211 \times 4 + 60,\\
211 &= 60 \times 3 + 31,\\
60 &= 31 \times 1 + 29,\\
31 &= 29 \times 1 + 2,\\
29 &= 2 \times 14 + 1,\\
2 &= 1 \times 2.
\end{aligned}$$

Por lo tanto, el máximo común divisor de 2019 y 1115 es $(2019, 1115) = 1$.

De las ecuaciones obtenidas arriba, es fácil ver que r_i es combinación lineal de los dos anteriores, $r_i = r_{i-2} - q_i r_{i-1}$. Más aún, de dichas relaciones se puede expresar a r_m como combinación lineal de cada par de residuos consecutivos r_{i-1} y r_i. Veámoslo como continuación del ejemplo anterior.

Ejemplo 2.5 *Del ejemplo 2.4, tenemos que*

$$\begin{aligned}
1 &= 29 - 2 \times 14,\\
2 &= 31 - 29 \times 1,\\
29 &= 60 - 31 \times 1,\\
31 &= 211 - 60 \times 3,\\
60 &= 904 - 211 \times 4,\\
211 &= 1115 - 904 \times 1,\\
904 &= 2019 - 1115 \times 1,
\end{aligned}$$

donde cada uno de los residuos ha sido expresado como combinación lineal de los anteriores. Usaremos dichas relaciones para expresar el último residuo, es decir, el 1, como combinación lineal de 2019 y 1115. En las siguientes igualdades,

$$\begin{aligned}
1 &= 29 \times 1 - 2 \times 14 = 29 - (31 - 29 \times 1) \times 14 \\
&= 29 \times 15 - 31 \times 14 = (60 - 31 \times 1) \times 15 - 31 \times 14 \\
&= 60 \times 15 - 31 \times 29 = 60 \times 15 - (211 - 60 \times 3) \times 29 \\
&= 60 \times 102 - 211 \times 29 = (904 - 211 \times 4) \times 102 - 211 \times 29 \\
&= 904 \times 102 - 211 \times 437 = 904 \times 102 - (1115 - 904 \times 1) \times 437 \\
&= 904 \times 539 - 1115 \times 437 = (2019 - 1115 \times 1) \times 539 - 1115 \times 437 \\
&= 2019 \times 539 - 1115 \times 976,
\end{aligned}$$

la columna de la izquierda presenta al 1 como combinación lineal de los otros residuos hasta llegar a 2019 y 1115, mientras que la columna de la derecha muestra el procedimiento que lleva a la siguiente combinación a partir de las expresiones de arriba. El lector puede verificar que, efectivamente, $2019 \times 539 - 1115 \times 976$ *es igual a 1.*

Proposición 2.3 (Algoritmo euclidiano extendido) *Sean a y n enteros positivos. El máximo común divisor $d = (a, n)$ puede expresarse como combinación lineal de a y n, es decir, $d = a \times b + n \times k$ para algunos enteros b y k. Más aún, los enteros b y k son únicos módulo n/d y a/d, respectivamente. En particular, si $d = 1$, entonces b es el inverso de a módulo n.*

Ejemplo 2.6 *Tomemos $n = 2019$ y $a = 1115$. Puesto que $2019 \times 539 - 1115 \times 976 = 1$, se concluye que $b = -976$ es el inverso multiplicativo de 1115 módulo 2019.*

Hemos visto que el inverso multiplicativo de $\mathbf{a} \in \mathbb{Z}/n\mathbb{Z}$ puede construirse si $(a, n) = 1$, aplicando el algoritmo euclidiano extendido para expresar a 1 como combinación lineal de a y n. Es importante mencionar que este procedimiento no sólo es válido para los enteros módulo n, es decir, al trabajar con clases de equivalencia del anillo de los enteros $\mathbb{Z}$, sino que también aplica en anillos más generales, llamados *euclidianos*, en los cuales se puede llevar a cabo una versión del algoritmo euclidiano arriba descrito. Esto es significativo, por ejemplo, para entender el trasfondo matemático del *Advanced Encryption Standard (AES)*, que es uno de los métodos de cifrado más utilizados actualmente, y parte del cual está basado en la construcción de inversos de clases de equivalencia de polinomios [12, Sección 4.3].

3. Cifrados de bloque: algunos ejemplos

3.1. ¿Por qué utilizar cifrados de bloque?

Antes de entrar de lleno con los cifrados de bloque, analicemos el cifrado de Julio César desde la perspectiva de la aritmética modular. Recordemos que este cifrado lo que hace es

recorrer cíclicamente el orden del alfabeto. Si a cada una de las letras a, b, c, ..., y, z le hacemos corresponder los números 0, 1, 2, ..., 25, 26, entonces la clave de la tabla 1 lo que hace es reemplazar el 0 por el 11, el 1 por el 12, el 2 por el 13, ..., el 15 por el 26, el 16 por el 0, el 17 por el 1, ... y el 26 por el 10.

Desde el punto de vista de la aritmética modular, el cifrado lo que hace es reemplazar la letra asociada con el número a con la letra asociada al número $a + 11 \mod 27$, es decir, al representante de clase módulo 27 de $a + 11$ que está entre 0 y 26. Esto lo podemos ver en la tabla 4, donde cada entrada del tercer renglón es el resultado de sumarle 11 en módulo 27 a la entrada que está arriba de ella.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	0	1	2	3	4	5	6	7	8	9	10
L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K

Tabla 4: La clave de la tabla 1 es simplemente sumar 11 en la artimética módulo 27.

Ejemplo 3.1 *El mensaje "este texto es indescifrable" usando números, y omitiendo los espacios en blanco, se expresaría por*

$$x = 4\ 19\ 20\ 4\ 20\ 4\ 24\ 20\ 15\ 4\ 19\ 8\ 13\ 3\ 4\ 19\ 2\ 8\ 5\ 18\ 0\ 1\ 11\ 4.$$

Sumando 11 $\mod 27$*, obtenemos*

$$y = 15\ 3\ 4\ 15\ 4\ 15\ 8\ 4\ 26\ 15\ 3\ 19\ 24\ 14\ 15\ 3\ 13\ 19\ 16\ 2\ 11\ 12\ 22\ 15.$$

Dichos números corresponden a las letras "ODEOEOIEZODSXÑODNSPCLMVO", que es el mensaje cifrado.

Con un alfabeto de 27 letras, solamente hay 27 opciones para crear un cifrado de Julio César: simplemente elegimos cuál número sumar módulo 27, que en este caso fue 11. Esto se expresa diciendo que el *espacio de claves* del cifrado de Julio César tiene solamente 27 elementos. Con la tecnología actual, este tamaño del espacio de claves no ofrece ninguna seguridad, tan sólo le serviría a los niños para enviar mensajes secretos a los amigos de la escuela.

El cifrado de Julio César es simplemente una *traslación* en módulo 27, pues cada uno de los números se suma por un elemento fijo. Un primer intento para construir un cifrado tal que el tamaño del espacio de claves ofrezca un mayor nivel de seguridad sería usando *transformaciones afines* módulo 27, es decir, combinar una transformación lineal con una traslación. En este caso, para el cifrado elegiríamos dos enteros a y b entre 0 y 26, tal que $(a, 27) = 1$. La cualidad de que a sea primo relativo con 27 es para que a tenga inverso, y que el mensaje cifrado pueda ser leído (descifrado). Luego, una vez que nuestro mensaje se convierte en una serie de números $x = x_1x_2 \dots x_n$ módulo 27, cada uno de los números x_i es transformado en $y_i = ax_i+b$. La persona que recibe el mensaje cifrado, para descifrarlo deberá

aplicar la transformación afín inversa $x_i := cy_i + d \mod 27$, donde $c := a^{-1}$ y $d := -a^{-1}b$. Es un ejercicio sencillo para el lector verificar que esto efectivamente define la transformación inversa del cifrado.

Ejemplo 3.2 *Sean $a = 2$ y $b = 3$. En este caso, el cifrado consiste en multiplicar por $a = 2$ y al resultado sumarle $b = 3$. Luego, la relación entre cada caracter con su cifrado sería la dada por la tabla 5 Nuevamente, si queremos cifrar el mensaje "este texto es indescifrable", pero ahora usando nuestra transformación afín $y = 2x + 3 \mod 27$, el resultado es "JMÑJÑJVÑEJMQAHJMFQLKBDWJ". La persona que recibe este mensaje tendría que usar la clave anterior pero en sentido inverso. Para encontrarla, la transformación afín a usar debe ser la inversa a $y = 2x+3 \mod 27$. Puesto que $14 \times 2 = 28 \equiv 1 \mod 27$, tenemos que 14 es el inverso de 2. Como $14 \times 3 \equiv 15 \mod 27$, resulta que la transformación afín inversa es $x = 14y - 15 \mod 27$.*

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
3	5	7	9	11	13	15	17	19	21	23	25	0	2	4	6	8	10	12	14	16	18	20	22	24	26	1
d	f	h	j	l	n	o	q	s	u	w	y	a	c	e	g	i	k	m	ñ	p	r	t	v	x	z	b

Tabla 5: La clave de la transformación afin $y_i = ax_i + b \mod 27$, con $a = 2$, $b = 3$.

Hemos pasado de usar traslaciones módulo 27 a transformaciones afines. En este caso, el espacio de clave consiste de $18 \times 27 = 486$ claves distintas, pues tenemos 18 opciones para elegir a tal que $(a, 27) = 1$ y 27 opciones para elegir b. Ahora el tamaño de clave es mucho más grande que el original de 27 que teníamos con el cifrado de Julio César, pero nuevamente este tamaño de clave es muy pequeño y no resistirá un ataque por *fuerza bruta* con las computadoras actuales. Un ataque por fuerza bruta consiste en probar cada una de las 486 posibles claves hasta encontrar la que funciona.

De la discusión anterior, vemos que es indispensable tener un espacio de claves bastante grande para poder resistir el ataque más elemental, que es el de prueba y error. Sin embargo, *no es suficiente* que el espacio de clave sea muy grande para asegurar que el cifrado sea seguro, como veremos en el siguiente ejemplo.

Las traslaciones y las transformaciones afines de caracter a caracter ofrecen un espacio de clave bastante pequeño. Una generalización es considerar *todas las posibles permutaciones* de nuestro alfabeto de 27 caracteres. Es decir, asociar a cada letra del alfabeto alguna otra sin algún orden aparente. Por ejemplo, la a con la R, la b con la X, la c con la H, la d con la E, etc. En este caso, el espacio de clave será de tamaño 27!, es decir, $27 \times 26 \times 25 \times \ldots \times 3 \times 2 \times 1 = 10888869450418352160768000000 \approx 1 \times 10^{28}$. Si contáramos con una potencia de cómputo tal que pudiéramos hacer un ataque por fuerza bruta capaz de probar un trillón de claves por segundo, nos podría tomar la edad actual del universo el encontrar la permutación que se utilizó para cifrar el mensaje. En este sentido, esta clase más amplia de cifrados es bastante resistente a ataques por fuerza bruta. Sin embargo, existen maneras más inteligentes de atacar a este cifrado, como veremos en el siguiente ejemplo.

Ejemplo 3.3 *Consideremos el mensaje cifrado*

```
ZKQ GDHGOTRQR KTETWQDOQ FZT ZK EQBGH XTEAHDOQY

RTST ATKTD GQDQ WTD UDQROTKAT TW FZT WZ

DHAQEOHKQY WTQ ETDH, GTDH TY DTEOGDHEH WHYQBTKAT

WT AOTKT TK DTUOHKTW WOBGYTBTKAT EHKTVQW.
```

en el que cada caracter se representa por una única letra. Los espacios entre palabras se han conservado, al igual que los signos de puntuación. Los acentos no se han tomado en cuenta.

T	Q	D	K	H	W	O	E	G	A	Z	Y	R	B	F	U	X	S	V
31	14	13	12	12	10	9	8	7	7	5	5	4	4	2	2	1	1	1

Tabla 6: Tabla de frecuencias del ejemplo 3.3.

Obsérvese en la tabla de frecuencias 6 que los caracteres más utilizados fueron `T`*,* `Q`*,* `D`*,* `K`*,* `H` *y* `W` *en ese orden. Tomando cuenta que en la lengua castellana los caracteres más utilizados en el lenguaje escrito son las letras "e", "a", "o", "s", "r" "n"(en ese orden), podemos empezar haciendo prueba y error cambiando los caracteres más utilizados del texto cifrado por algunos de estos últimos. Por ejemplo, cambiando* `T` *por* `e` *y* `Q` *por* `a`*, obtenemos*

```
ZKa GDHGOeRaR KeEeWaDOa FZe ZK EaBGH XeEAHDOaY

ReSe AeKeD GaDa WeD UDaROeKAe eW FZe WZ

DHAaEOHKaY Wea EeDH, GeDH eY DeEOGDHEH WHYaBeKAe

We AOeKe eK DeUOHKeW WOBGYeBeKAe EHKeVaW.
```

Pensando en que también la letra `W` *fue una de las más utilizadas, podemos suponer que se trate de la* `o`*, la* `s`*, la* `r` *o la* `n`*. Haciendo la prueba con cada una de ellas, vemos que lo que mejor corresponde a lo que ya hemos descifrado es sustituirla por la* `s`*:*

```
ZKa GDHGOeRaR KeEesaDOa FZe ZK EaBGH XeEAHDOaY

ReSe AeKeD GaDa seD UDaROeKAe es FZe sZ
```

```
DHAaEOHKaY sea EeDH, GeDH eY DeEOGDHEH sHYaBeKAe

se AOeKe eK DeUOHKes sOBGYeBeKAe EHKeVas.
```

Tratando de reemplazar `D`*,* `K` *y* `H` *por* `o`*,* `r` *y* `n` *en algún orden, puede verse que la* `D` *y la* `K` *no corresponden a la* `o` *y que probablemente* `D` *corresponda a* `r`*. Así, reemplazando* `H` *por* `o`*,* `D` *por* `r` *y* `K` *por* `n`*, obtenemos*

```
Zna GroGOeRaR neEesarOa FZe Zn EaBGo XeEAorOaY

ReSe Aener Gara ser UraROenAe es FZe sZ

roAaEOonaY sea Eero, Gero eY reEOGroEo soYaBenAe

se AOene en reUOones sOBGYeBenAe EoneVas.
```

De este mensaje se puede deducir que la `Z` *corresponde a* `u` *y que la* `E` *representa* `c`*. Haciendo ese cambio, se puede deducir que* `O` *representa* `i` *y después que* `A` *representa* `t`*. Después de unas cuantas pruebas y errores, se puede descifrar el mensaje:*

```
una propiedad necesaria que un campo vectorial

debe tener para ser gradiente es que su

rotacional sea cero, pero el reciproco solamente

se tiene en regiones simplemente conexas.
```

En el ejemplo anterior se explotó la siguiente debilidad del cifrado: el mensaje cifrado conserva las *propiedades estadísticas* del mensaje original [12, Subsección 1.2.2]. Esto permitió que con un poco de deducción lógica y basándonos en reglas de ortografía elementales [4], se pudiera deducir el contenido del mensaje en un tiempo relativamente corto (menor que la edad del universo).

Una lección que nos brinda el ejemplo anterior es que aunque el tener un espacio de claves grande es una condición necesaria para que un cifrado resista ataques por fuerza bruta, esto no es suficiente para garantizar que el cifrado sea seguro. En virtud de ello, conviene presentar un par de propiedades adicionales que se necesitan tener para que un cifrado sea seguro [12, Subsección 3.1.1]:

1. **Confusión:** La relación entre el mensaje cifrado y la clave utilizada para cifrarlo no es inmediata.

2. **Difusión:** El cambiar un símbolo del mensaje original influye en muchos de los símbolos del texto cifrado.

El cifrado por sustitución que presentamos arriba posee la propiedad de confusión, pero no la de difusión. Es importante mencionar que para que un cifrado sea seguro, se necesita que se tengan ambas propiedades al mismo tiempo. En la práctica, los cifrados de bloque aplican alternadamente operaciones de confusión y difusión para una mayor seguridad, es el caso del DES y el AES. En la siguiente parte presentaremos un primer ejemplo de cifrado de bloque, el cual lo podemos pensar como un procedimiento para obtener confusión y difusión (pero que tampoco es lo bastante seguro por sí solo, como veremos en su momento).

Para evitar que un cifrado conserve las propiedades estadísticas del mensaje original, debemos evitar operar caracter a caracter. Una manera de hacerlo es utilizando *cifrados de bloque*, como veremos a continuación.

3.2. Transformaciones afines

El cifrado que presentamos en este apartado consiste en aplicar transformaciones afines, pero a diferencia de las que presentamos arriba y que sólo nos permitían transformar un caracter a la vez, éstas utilizan multiplicación de matrices y suma de vectores, de una cierta longitud fija. Esto nos permitirá cifrar *bloques completos* de dicha longitud fija de una sola vez.

En este procedimiento, nuevamente pensaremos en las 27 letras del abecedario, junto con el espacio en blanco, como números de 0 a 27, o más precisamente, como *clases de residuos módulo 28* (ver tabla 7). Naturalmente, este procedimiento puede adaptarse para admitir un mayor número de caracteres como letras con acentos, espacios y signos de puntuación. Una manera de hacer eso es mediante el *código ASCII*[1], que permite manejar 256 caracteres distintos.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	ñ	o	p	q	r	s	t	u	v	w	x	y	z	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

Tabla 7: La relación que permite trabajar numéricamente con letras.

Cifrando bloques de longitud 3. Para fines ilustrativos, vamos a presentar una manera de cifrar bloques de longitud 3, para lo cual utilizaremos la siguiente matriz de tamaño 3×3 y el vector de longitud 3:

[1]El código ASCII puede consultarse en `https://elcodigoascii.com.ar/`.

$$A = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

En general, para cifrar bloques de longitud n se tomaría una matriz $n \times n$ y un vector de tamaño n. Lo único que debemos de cuidar es que la matriz A que elijamos sea invertible módulo 28, para poder descifrar correctamente el mensaje. Esto quiere decir que el *determinante* de A sea primo relativo con 28, $(\det A, 28) = 1$.

Utilizando A y b podemos definir una transformación afín $y = Ax + b$, donde x y y son bloques de longitud 3. Por ejemplo, supongamos que queremos cifrar el mensaje "ejemplos de cifrados de bloque". Para lograrlo, necesitamos dividir nuestro mensaje en bloques verticales de tamaño 3, incluyendo los espacios:

$$\begin{pmatrix} e & m & o & d & c & r & o & d & b & q \\ j & p & s & e & i & a & s & e & l & u \\ e & l & & & f & d & & & o & e \end{pmatrix}.$$

Utilizando la relación de la tabla 7, podemos convertir nuestro arreglo de letras en una matriz X de tamaño 3×10 con entradas en $\mathbb{Z}/28\mathbb{Z}$,

$$X = \begin{pmatrix} 4 & 12 & 15 & 3 & 2 & 18 & 15 & 3 & 1 & 17 \\ 9 & 16 & 19 & 4 & 8 & 0 & 19 & 4 & 11 & 21 \\ 4 & 11 & 27 & 27 & 5 & 3 & 27 & 27 & 15 & 4 \end{pmatrix}.$$

Ahora bien, para obtener el mensaje cifrado, a cada columna x_i le vamos a aplicar la transformación afín mencionada arriba para obtener el bloque cifrado $y_i = Ax_i + b$. Es decir, multiplicamos a la matriz del mensaje X por la matriz A, y a cada columna de la matriz resultante le sumamos el vector b. Cabe mencionar que las operaciones de suma y multiplicación se efectúan módulo 28:

$$\begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 & 12 & 15 & 3 & 2 & 18 & 15 & 3 & 1 & 17 \\ 9 & 16 & 19 & 4 & 8 & 0 & 19 & 4 & 11 & 21 \\ 4 & 11 & 27 & 27 & 5 & 3 & 27 & 27 & 15 & 4 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 22 & 15 & 9 & 7 & 21 & 3 & 9 & 7 & 9 & 18 \\ 12 & 6 & 13 & 1 & 12 & 24 & 13 & 1 & 3 & 25 \\ 17 & 12 & 21 & 10 & 12 & 8 & 21 & 10 & 13 & 27 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 23 & 16 & 10 & 8 & 22 & 4 & 10 & 8 & 10 & 19 \\ 11 & 5 & 12 & 0 & 11 & 23 & 12 & 0 & 2 & 24 \\ 17 & 12 & 21 & 10 & 12 & 8 & 21 & 10 & 13 & 27 \end{pmatrix}$$

Esta matriz resultante la volvemos a converitr a letras, si así se prefiere, de acuerdo a la tabla 7. Así, obtenemos el mensaje cifrado "WLQPFMKMUIAKVLMEWIKMUIAKKCNSX ", donde al final hay un espacio.

Una de las ventajas que tiene el cifrar de esta manera es que un mismo caracter no corresponde siempre al mismo caracter. En efecto, la palabra *ejemplos* se convirtió en $WLQPFMKM$ donde las letras e del mensaje original fueron a parar a letras distintas, W y Q. Por otra parte, dos letras distintas l y s fueron a parar al mismo caracter. Lo que de fondo está ocurriendo es que nuestro cifrado toma bloques completos de longitud 3 y cifra su contenido. Por ejemplo, en nuestro mensaje original aparece dos veces el bloque "os " (con espacio al final), el cual se cifra como "KMU" las dos veces que aparece. Lo mismo ocurre con el bloque "de ".

Descifrando el mensaje. Para que el receptor del mensaje cifrado pueda leerlo, necesitará descifrarlo. En este caso, hay que usar una *transformación afín inversa*. Recordemos que cada bloque del mensaje original x de longitud 3 fue cifrado con una transformación afín $y = Ax+b$. Para recuperar x a partir de y, usaremos la transformación afín $x = Cy + d$, donde $C = A^{-1}$ y $d = -A^{-1}b$. En nuestro caso,

$$C = A^{-1} = \begin{pmatrix} 6 & 25 & 16 \\ 16 & 6 & 25 \\ 25 & 16 & 6 \end{pmatrix} \qquad d = -A^{-1}b = \begin{pmatrix} 19 \\ 18 \\ 19 \end{pmatrix}.$$

El lector puede verificar que al tomar cada bloque y, de longitud 3 en el mensaje cifrado, y aplicarle la transformación afín inversa $x = Cy + d$, recuperamos la matriz del mensaje original.

La debilidad de la linealidad. Notemos que cada uno de los tres caracteres que conforman a un bloque cifrado y_i dependen de los tres caracteres del bloque original x_i. En este sentido, nuestro cifrado de bloque tiene buena difusión dentro del bloque de tamaño 3. A pesar de esto, este modo en que estamos implementando las transformaciones afines *sí conserva algunas propiedades estadísticas del mensaje original* y en principio sí pudiera explotarse dicha debilidad para descifrar el contenido del mensaje. Sin embargo, la mayor debilidad que presenta este cifrado se relaciona con la inherente linealidad que lo define. Esto hace que *con muy poca información que se conozca, se pueda descubrir la clave* con que se cifra un mensaje. En efecto, puesto que nuestro cifrado utiliza transformaciones afines de orden 3, a un atacante le puede bastar conocer cuatro parejas de bloques original-cifrado (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) para descubrir la clave. Más precisamente, es suficiente que $v_1 := x_1 - x_0$, $v_2 := x_2 - x_0$ y $v_3 := x_3 - x_0$ sean linealmente independientes para recuperar la transformación afín. Si denotamos $w_1 := y_1 - y_0$, $w_2 := y_2 - y_0$ y $w_3 := y_3 - y_0$, entonces

$$w_i = y_i - y_0 = (Ax_i + b) - (Ax_0 + b) = A(x_i - x_0) = Av_i.$$

Por lo tanto, para conocer A, basta resolver el sistema lineal $Av_1 = w_1$, $Av_2 = w_2$ y $Av_3 = w_3$.

Supongamos que un atacante ha logrado interceptar parte del mensaje anterior, por ejemplo, que sabe que la palabra "*ejemplos de* "(con espacio al final) ha sido cifrado en "WLQPFMKMUIAK". Con esta información, podemos recuperar la matriz A utilizada

arriba. En nuestro caso, tenemos que

$$x_0 = \begin{pmatrix} e \\ j \\ e \end{pmatrix} = \begin{pmatrix} 4 \\ 9 \\ 4 \end{pmatrix}, \quad x_1 = \begin{pmatrix} m \\ p \\ l \end{pmatrix} = \begin{pmatrix} 12 \\ 16 \\ 11 \end{pmatrix}, \quad x_2 = \begin{pmatrix} o \\ s \end{pmatrix} = \begin{pmatrix} 15 \\ 19 \\ 27 \end{pmatrix}, \quad x_3 = \begin{pmatrix} d \\ e \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 27 \end{pmatrix},$$

por lo cual

$$v_1 = x_1 - x_0 = \begin{pmatrix} 8 \\ 7 \\ 7 \end{pmatrix}, \qquad v_2 = x_2 - x_0 = \begin{pmatrix} 11 \\ 10 \\ 23 \end{pmatrix}, \qquad v_3 = x_3 - x_0 = \begin{pmatrix} 27 \\ 23 \\ 23 \end{pmatrix}.$$

Similarmente,

$$y_0 = \begin{pmatrix} W \\ L \\ Q \end{pmatrix} = \begin{pmatrix} 23 \\ 11 \\ 17 \end{pmatrix}, \quad y_1 = \begin{pmatrix} P \\ F \\ M \end{pmatrix} = \begin{pmatrix} 16 \\ 5 \\ 12 \end{pmatrix}, \quad y_2 = \begin{pmatrix} K \\ M \\ U \end{pmatrix} = \begin{pmatrix} 10 \\ 12 \\ 21 \end{pmatrix}, \quad y_3 = \begin{pmatrix} I \\ A \\ K \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 10 \end{pmatrix},$$

por lo cual

$$w_1 = y_1 - y_0 = \begin{pmatrix} 21 \\ 22 \\ 23 \end{pmatrix}, \qquad w_2 = y_2 - y_0 = \begin{pmatrix} 15 \\ 1 \\ 4 \end{pmatrix}, \qquad w_3 = y_3 - y_0 = \begin{pmatrix} 13 \\ 17 \\ 21 \end{pmatrix}.$$

Luego, aplicando Gauss-Jordan con los vectores traspuestos de v_1, v_2, v_3 y w_1, w_2, w_3,

$$\left(\begin{array}{ccc|ccc} 8 & 7 & 7 & 21 & 22 & 23 \\ 11 & 10 & 23 & 15 & 1 & 4 \\ 27 & 23 & 23 & 13 & 17 & 21 \end{array}\right),$$

podemos recuperar la traspuesta de la matriz A. En efecto, primero multiplicamos al primer renglón por 23 y al resultado le restamos 7 veces el tercero:

$$\left(\begin{array}{ccc|ccc} 23 & 0 & 0 & 0 & 23 & 18 \\ 11 & 10 & 23 & 15 & 1 & 4 \\ 27 & 23 & 23 & 13 & 17 & 21 \end{array}\right).$$

Observemos que el inverso de 23 módulo 28 es 11. Luego, multiplicando al primer renglón por 11, obtenemos

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 2 \\ 11 & 10 & 23 & 15 & 1 & 4 \\ 27 & 23 & 23 & 13 & 17 & 21 \end{array}\right)$$

En este paso hemos obtenido que (0 1 2) es el primer renglón de la matriz traspuesta de A. Procediendo de manera similar, recuperamos la matriz A que usamos para cifrar el mensaje. Finalmente, el vector b se recupera haciendo

$$b = y_0 - Ax_0 = \begin{pmatrix} 23 \\ 11 \\ 17 \end{pmatrix} - \begin{pmatrix} 0 & 2 & 1 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 9 \\ 4 \end{pmatrix} = \begin{pmatrix} 23 \\ 11 \\ 17 \end{pmatrix} - \begin{pmatrix} 22 \\ 12 \\ 17 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

En general, cuando al cifrar se usa una transformación afín de orden n, es suficiente tener $n+1$ parejas *genéricas* $(x_0, y_0), \ldots, (x_n, y_n)$ para poder descifrarlo, lo cual en términos prácticos representa un nivel de seguridad demasiado bajo.

3.3. El telegrama Zimmermann

Hace poco más de un siglo, Europa estaba en medio del peor conflicto armado hasta entonces: la Primera Guerra Mundial. Hacia inicios de 1917, Estados Unidos aún se mantenía neutral. Sin embargo, mantenía fuertes relaciones comerciales con Francia y el Imperio Británico. Particularmente, los banqueros estadounidenses habían realizado fuertes préstamos a dichos países para que pudieran sostenerse durante la guerra. En consecuencia, Estados Unidos tenía interés en que estos países resultaran vencedores en este conflicto.

En el bando contrario, el Imperio Alemán utilizaba sus submarinos para atacar los barcos comerciales de Estados Unidos que viajaban rumbo a Inglaterra. Aún así, buscaban que los Estados Unidos no se involucraran directamente dentro del conflicto armado. Sin embargo, la entrada de los Estados Unidos en la primera guerra mundial era un hecho cada vez más inevitable. En este conexto, el ministro Alemán de Asuntos Exteriores Arthur Zimmermann envió un telegrama el 16 de enero de 1917 a su embajador en México, el conde Heinrich von Eckardt. En dicho telegrama, le daba instrucciones de que, en caso de que Estados Unidos decidiera entrar al conflicto armado, formara una alianza militar con México, a cambio de la cual se le ofrecería devolverle una parte de los territorios que México había perdido 69 años antes cuando los Estados Unidos invadieron el país.

Por su parte, el gobierno de México, encabezado entonces por Venustiano Carranza, declinó la oferta. De acuerdo con [9], cuando el contenido del telegrama fue hecho público, el gobierno mexicano negó haberlo recibido, aunque hubo testigos asegurando que Carranza y sus colaboradores cercanos recibieron y rechazaron inmediatamente la propuesta, mientras que también hay quienes afirman que la propuesta no fue entregada a Carranza por lo peligroso de la misma. En cualquier caso, la propuesta en sí misma era inviable desde el punto de vista mexicano. En esa época, el gobierno estaba ocupado socabando las revueltas de Pancho Villa en el norte y de Emiliano Zapata en el sur. Asimismo, la fuerza militar de los Estados Unidos era ya bastante más poderosa que la de México, como se había exhibido pocos años antes en la ocupación estadounidense de Veracruz en 1914.

El telegrama Zimmermann fue enviado de forma cifrada, tal como se muestra en la figura 2. Resulta que el mismo día en que el telegrama fue enviado, los ingleses lo interceptaron y descifraron, gracias a que ya conocían parte del cifrado utilizado. El método de cifrado utilizado era una especie de diccionario, es decir, cada uno de los bloques numéricos del telegrama representa una palabra. Por ejemplo, algunas de las palabras que aparecen en dicho telegrama son [5, Capítulo I]:

14936	eingeschränkten	22049	sich
15021	einzeln	22200	stop
15099	Empfang	22295	sofortiger

Al descifrado del telegrama Zimmermann, y su revelación al gobierno estadounidense por parte de los ingleses, se le ha llamado “el mayor golpe de inteligencia de todos los tiempos” [5, Capítulo I]. Esto en parte se debe a que, al descifrar y publicar su contenido, se logró cambiar la postura antibelicista de muchos estadounidenses, siendo el último empujón para la entrada de los Estados Unidos en la guerra, favoreciendo decisivamente el fin del conflicto.

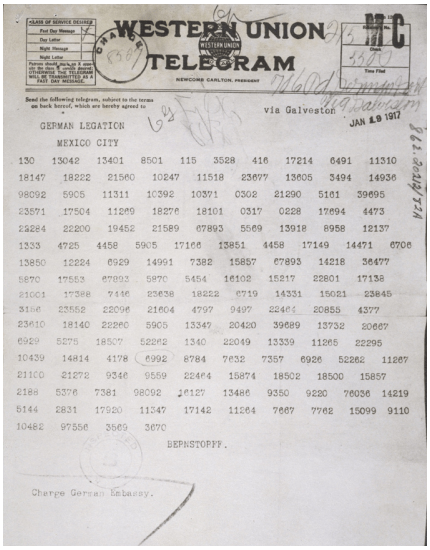
WESTERN UNION
TELEGRAM
via Galveston
JAN 19 1917
GERMAN LEGATION
MEXICO CITY
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670
BEPNSTOPFF.
Charge German Embassy.

Figura 2: El telegrama Zimmermann.

4. Data Encryption Standard

Los ejemplos de los apartados anteriores nos han permitido ilustrar algunas de las cualidades que son deseables en los cifrados, así como ciertas debilidades que conviene evitar. Por un lado, la propiedad de confusión nos ayuda a que no sea fácil identificar la clave con la que se ha cifrado el mensaje. La propiedad de difusión, por otro lado, ayuda a que pequeños cambios en el mensaje original se traduzcan en varios cambios al mensaje cifrado. Finalmente, hemos visto que la linealidad en los cifrados brinda difusión, pero es muy fácil de descifrar conociendo poca información.

A continuación, presentaremos el *Data Encryption Standard (DES)*, el cual cifra bloques de 64 bits. Quiere decir que, en lugar de trabajar con bloques formados por enteros del 0 al 27 como en la correspondencia de la tabla 7, *los bloques a cifrar consisten únicamente de ceros y unos*. En resumen, el *DES* cifra bloques de ceros y unos de longitud 64.

Históricamente, los cifrados utilizados por los gobiernos se habían manejado de manera secreta. O sea que no se conocía tanto el procedimiento de *cifrado* en sí como la *clave* usada por dicho cifrado. Hoy en día, los esquemas de cifrado utilizados son conocidos. Es decir, se sabe cuál es el cifrado utilizado por las instituciones gubernamentales, financieras o de comunicación, manteniendo oculta la clave que se utiliza, por supuesto. Esto tiene sus grandes ventajas, pues al ser conocido o público el esquema de cifrado que se utiliza, hay muchos más criptoanalistas trabajando para estudiar y descubrir las posibles vulnerabilidades del cifrado. De esta manera, se puede confiar más en la seguridad propia del cifrado que en la secrecía del mismo.

Este paso de mantener los cifrados en secreto a hacerlos públicos se dio justamente con el *DES* [12, Capítulo 3]. En 1972 el US National Bureau of Standards (NBS), hoy llamado *National Institute of Standards and Technology*, realizó solicitudes para un cifrado de uso estandarizado en los Estados Unidos. Se buscaba que dicho cifrado pudiera utilizarse en

diferentes aplicaciones, tanto de gobierno, como financieras y comerciales. Dos años más tarde, recibieron una propuesta de un grupo de criptógrafos de IBM. Dicha propuesta está basada en un algoritmo conocido como *red de Feistel*, que describiremos en su momento. El cifrado propuesto se llamaba *Lucifer* (en inglés, *-cifer* se pronuncia igual que la palabra *cipher*, que significa "cifrado"), el cual tenía la capacidad de cifrar bloques de 64 bits usando una clave que consistía de 128 bits.

La NBS remitió la examinación de la seguridad del cifrado propuesto a la *National Security Agency* (NSA), cuya existencia no era admitida en aquella época. Dicha agencia de seguridad, además de cambiar el nombre del cifrado a *Data Ecryption Standard*, decidió reducir el tamaño de clave de 128 a 56 bits, haciéndolo más vulnerable a ataques por fuerza bruta. Debido a esto, se temía que dicha agencia hubiera encontrado alguna vulnerabilidad matemática sólo conocida por ellos que les permitiera romper el cifrado a voluntad. A pesar de esas inquietudes, las especificaciones del cifrado fueron puestas a disposición del público en 1977. El haber hecho público el algoritmo de dicho cifrado, junto con el rápido crecimiento en el uso de computadoras a principos de los ochenta, permitió que la comunidad de investigadores pudiera analizar a profundidad al *DES*.

4.1. Descripción general del cifrado

Como mencionamos arriba, el *DES* cifra bloques de 64 bits usando claves de 56 bits basado en una *red de Feistel*. Esto significa que al principio se aplica al bloque de 64 bits una permutación inicial; después se aplica un algoritmo iterativo de 16 rondas, en cada una de las cuales se realiza prácticamente el mismo procedimiento. Por último, se aplica una permutación final. A partir de la clave original k se derivan 16 subclaves $k_1, k_2, \ldots, k_{16}$, cada una de las cuales se utiliza en cada ronda. Este procedimiento puede visualizarse esquemáticamente en el diagrama de la figura 3.

Una descripción un poco más detallada del procedimiento es la siguiente. Primeramente, al mensaje original x, que es un bloque de 64 bits, se le aplica una *permutación inicial* $IP(x)$. El bloque resultante es dividido en dos bloques L_0 y R_0 (izquierdo y derecho) de 32 bits cada uno. Ambas mitades entran como argumento de la red de Feistel de 16 rondas. Para cada $i = 1, \ldots, 16$, al bloque (L_{i-1}, R_{i-1}) se le aplica el procedimiento de la i-ésima ronda, dando como resultado el bloque (L_i, R_i) por la fórmula siguiente:

$$L_i := R_{i-1}, \qquad R_i := L_{i-1} \oplus f(R_{i-1}, k_i).$$

Aquí, f es una función que toma el bloque derecho anterior R_{i-1} y la subclave de la ronda k_i y devuelve un bloque de longitud de 32 bits. La operación $\oplus$ es la disyunción exclusiva o XOR, que fue descrita en el ejemplo 2.3. Después de la ronda 16, se aplica la permutación final, que consiste en intercambiar las dos mitades del bloque (L_{16}, R_{16}) y aplicarle la inversa de la permutación inicial. El resultado es el mensaje cifrado,

$$y = \mathrm{DES}_k(x) = IP^{-1}(R_{16}, L_{16}).$$

En este procedimiento iterativo, las propiedades de confusión y difusión se dan en cada ronda. La propiedad de confusión es asegurada por la estructura de la función f: su im-

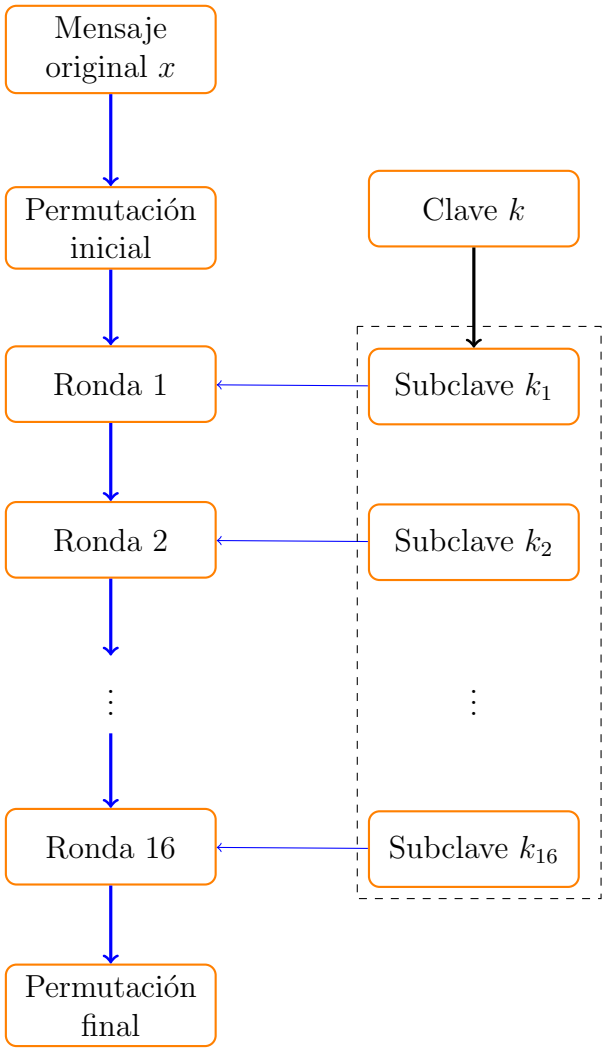

Figura 3: Diagrama de flujo del cifrado DES.

predictibilidad asegura que no exista una relación inmediata entre el mensaje cifrado y la clave original. Sin embargo, como puede observarse, de una ronda a otra solamente se cifra la mitad izquierda L_{i-1} cuando se suma con el resultado de la función f, mientras que la mitad derecha R_{i-1} pasa sin cifrarse a ser la nueva parte izquierda. Esto exige que para tener una alta difusión se deban de utilizar 16 rondas. El procedimiento que se aplica en cada ronda viene esquemáticamente descrito en la figura 4.

Para comprender a detalle el cifrado DES, necesitamos describir las permutaciones IP e IP^{-1}, explicar cómo opera la función f y cómo se generan las subclaves k_i a partir de la clave k.

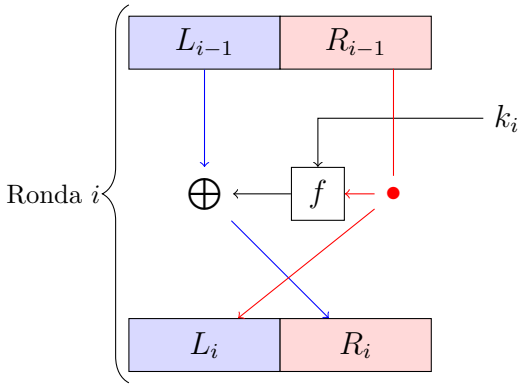

Figura 4: En cada ronda, el texto de 64 bits se divide en dos partes izquierda y derecha de 32 bits. La mitad derecha entra como argumento a la función f, pero pasa sin cifrarse a la parte izquierda. La mitad izquierda es cifrada cuando se suma con el resultado de la función f.

4.2. La permutación inicial y su inversa

La manera más concisa para describir la permutación inicial es la siguiente: Si $a = a_1a_2\ldots a_{64}$ es un bloque de 64 bits, entonces su imagen bajo la permutación inicial IP es $b = IP(a) := b_1b_2\ldots b_{64}$, donde para cada $i = 1,\ldots,64$, se define $b_n := a_{p(n)}$, con

$$p(n) := \begin{cases} 58n \mod 66 & \text{si } n \leq 32, \\ 58n - 9 \mod 66 & \text{si } n \geq 33. \end{cases}$$

Esto quiere decir que el primer bit del bloque resultante será el que estaba en la posición $p(1) = 58$, el segundo bit del bloque resultante será el que estaba en la posición $p(2) = 50$, y así sucesivamente. Si bien ésta es una forma concisa de describirla, para implementarla es más eficiente presentar todos los valores como en la tabla de la figura 5.

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Figura 5: La permutación inicial del DES. Tomada de [12, Tabla 3.1].

Similarmente, la permutación inversa IP^{-1} puede describirse de manera concisa como sigue: Si $b = b_1b_2\ldots b_{64}$ es un bloque de 64 bits, su imagen es $a = IP^{-1}(b) := a_1a_2\ldots a_{64}$, donde $a_m := b_{q(m)}$, con

$$q(m) := \begin{cases} 4m \mod 33 & \text{si } m \text{ es par}, \\ (4m+3 \mod 33) + 33 & \text{si } m \text{ es impar}. \end{cases}$$

Por ejemplo, el primer bit del bloque resultante será aquél que estaba en la posición $q(1) = 40$, el segundo bit resultante es el que estaba en la posición $q(2) = 8$, etcétera. Dicha permutación puede describirse por la figura 6.

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Figura 6: La permutación inversa a la inicial en el DES. Tomada de [12, Tabla 3.2].

4.3. La función *f*

La función f que aparece en el algoritmo del DES es el elemento esencial de este cifrado. Es la función f la que le otorga la propiedad de confusión. Asimismo, es el único elemento no lineal en el cifrado.

En cada ronda, esta función toma como argumentos a la subclave de la ronda y a la mitad derecha del bloque obtenido en la ronda anterior, de longitudes 48 y 32 bits, respectivamente, y devuelve un bloque de 32 bits. Más precisamente, en la ronda i la función f toma el bloque R_{i-1} y le aplica una *expansión* para convertirlo en un bloque $E(R_{i-1})$ de 48 bits. Posteriormente, el bloque expandido se suma, con la operación XOR, con la subclave k_i. El bloque resultante, de 48 bits, es dividido en ocho sub-bloques de 6 bits, cada uno de los cuales pasa por unas *cajas de sustitución*, que devuelven en total ocho bloques de 4 bits. Finalmente, el bloque completo de 32 bits es permutado, y el bloque obtenido es el resultado de la función, $f(R_{i-1}, k_i)$ (ver figura 7).

A continuación, describimos cada paso con detalle.

La expansión. Como comentábamos arriba, el primer paso que realiza la función es expandir el bloque R_{i-1}, de 32 bits, a otro bloque $E(R_{i-1})$ de 48 bits. Básicamente, lo que hace la función de expansión es repetir los bits que se ubican en las posiciones 0 y 1 módulo 4, es decir, los que están en las posiciones 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32 y 1. Esta expansión puede describirse por la figura 8. Dicha expansión, también puede describirse de manera concisa usando aritmética modular, tal como lo hicimos con la permutación incial y su inversa, sólo que usando una expresión un poco más complicada. Si $R = r_1 r_2 \dots r_{32}$ es el bloque de 32 bits a expandir, el resultado de la expansión es $E(r) := e_1 e_2 \dots e_{48}$, donde $e_n := r_{g(n)}$, con

$$g(n) := ((6n + 25 - 5(n - 1 \mod 6)) \mod 32) + 1, \qquad n = 1, 2, \dots, 48.$$

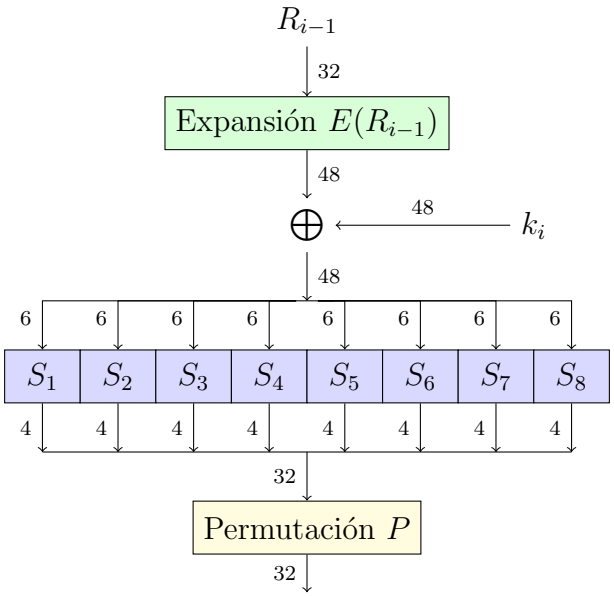

Figura 7: Descripción esquemática de la función f. A lado de cada flecha hemos indicado el tamaño de bloque correspondiente.

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Figura 8: La la expansión en la función f del DES. Tomada de [12, Tabla 3.3].

Las cajas de sustitución. Como se ha indicado en el diagrama de la figura 7, el bloque obtenido después de la expansión se suma con la subclave de la ronda. El bloque resultante, de 48 bits, es dividido en ocho sub-bloques de 6 bits cada uno, los cuales pasan a una de las ocho *cajas de sustitución*, o más brevemente, *S-cajas*.

Cada una de las S-cajas recibe un bloque de 6 bits y devuelve uno de 4 bits. Las S-cajas son las operaciones más importante del DES, en cuanto a que en ellas recae la seguridad del cifrado [3]. En efecto, las S-cajas son el único elemento no lineal de dicho algoritmo [5, Sección 2.2] y le brindan la propiedad de confusión [12, Subsección 3.3.2], evitando explotar en el DES las debilidades de la linealidad discutidas en la sección anterior.

A continuación, explicamos cómo es que dichas S-cajas están definidas y posteriormente haremos comentarios al respecto de dicha definición y su importancia.

Una manera de pensar en una S-caja es como una colección de cuatro permutaciones σ_0,

σ_1, σ_2, σ_3 del conjunto $\{0, 1, 2, \ldots, 14, 15\}$, las cuales difieren de una S-caja a otra. Cuando una S-caja recibe como argumento un bloque de 6 bits $b = b_5b_4b_3b_2b_1b_0$, el primer y último bit se piensan como la representación binaria de un entero x entre 0 y 3, $x := (b_5b_0)_2$. Similarmente, los bits restantes se piensan como la representación binaria de un entero y entre 0 y 15, $y := (b_4b_3b_2b_1)_2$. Los cuatro bits que la S-caja arroja como resultado son la representación binaria del resultado de aplicar la permutación σ_x de la S-caja al entero y: $S(b) := \sigma_x(y)$.

Para ilustrar la descripción general anterior, consideremos la primera S-caja del DES, la cual consiste de las permutaciones

$$\begin{aligned}
\sigma_0 &= (0\ 14)(1\ 4\ 2\ 13\ 9\ 10\ 6\ 11\ 12\ 5\ 15\ 7\ 8\ 3),\\
\sigma_1 &= (0)(1\ 15\ 8\ 10\ 12\ 9\ 6\ 13\ 5\ 2\ 7)(3\ 4\ 14)(11),\\
\sigma_2 &= (0\ 4\ 13\ 10\ 9\ 12\ 3\ 8\ 15)(1)(2\ 14\ 5\ 6)(7\ 11),\\
\sigma_3 &= (0\ 15\ 13)(1\ 12\ 10\ 3\ 2\ 8\ 5\ 9\ 11\ 14\ 6)(4)(7).
\end{aligned}$$

Hemos presentado cada permutación en términos de su descomposición en producto de ciclos ajenos. Esta notación significa, por ejemplo, que la permutación σ_3 envía el 0 al 15, el 15 al 13 y éste 13 de vuelta al 0; envía el 1 al 12, éste al 10, éste al 3, éste al 2, éste al 8, éste al 5, éste al 9, éste al 11, éste al 14, éste al 6 y éste de vuelta al 1; finalmente, el 4 es enviado en sí mismo, al igual que el 7.

Supongamos que la S-caja S_1 recibe por argumento al bloque de seis bits $b = 111010$. El entero x es aquél cuya representación binaria es 10, ya que estos son el primer y último bits de b. En otras palabras, es $x = 10_2 = 2$. Similarmente, es $y = 13$, pues los cuatro bits centrales de b, a saber, 1101, representan al número 13. Por tanto, debemos aplicar la permutación indicada por $x = 2$ al entero $y = 13$, es decir, $\sigma_2(13) = 10$. Finalmente, el resultado obtenido de aplicar la S-caja S_1 al bit b es la representación en cuatro bits (binaria) de 10, que es 1010:

$$S_1(111010) = \sigma_2(13) = 10 \equiv 1010_2.$$

Cada una de las ocho cajas de sustitución operan de manera análoga que S_1, sólo que las cuatro permutaciones de cada S-caja son diferentes:

- Permutaciones de la caja de sustitución S_2:

$$\begin{aligned}
\sigma_0 &= (0\ 15\ 10\ 2\ 8\ 9\ 7\ 4\ 6\ 3\ 14\ 5\ 11\ 13)(12),\\
\sigma_1 &= (0\ 3\ 7\ 14\ 11\ 10\ 1\ 13\ 9)(2\ 4\ 15\ 5)(6\ 8\ 12),\\
\sigma_2 &= (0)(1\ 14\ 2\ 7)(3\ 11\ 6\ 13)(4\ 10\ 12\ 9\ 8\ 5)(15),\\
\sigma_3 &= (0\ 13\ 5\ 15\ 9\ 6\ 4\ 3\ 1\ 8\ 11\ 12)(2\ 10\ 7)(14).
\end{aligned}$$

- Permutaciones de la caja de sustitución S_3:

$$\begin{aligned}
\sigma_0 &= (0\ 10\ 12\ 11\ 7\ 5\ 3\ 14\ 2\ 9\ 13\ 4\ 6\ 15\ 8\ 1),\\
\sigma_1 &= (0\ 13\ 11\ 14\ 15\ 1\ 7\ 10\ 5\ 4\ 3\ 9\ 8\ 2)(6)(12),\\
\sigma_2 &= (0\ 13\ 10\ 2\ 4\ 8\ 11\ 12\ 5\ 15\ 7)(1\ 6\ 3\ 9)(14),\\
\sigma_3 &= (0\ 1\ 10\ 14\ 2\ 13\ 5\ 9\ 15\ 12\ 11\ 3)(4\ 6\ 8)(7).
\end{aligned}$$

- Permutaciones de la caja de sustitución S_4:

$$\begin{aligned}\sigma_0 &= (0\ 7\ 10\ 8\ 1\ 13\ 12\ 11\ 5\ 6\ 9\ 2\ 14\ 4)(3)(15),\\ \sigma_1 &= (0\ 13\ 10\ 2\ 11\ 12\ 1\ 8\ 4\ 6)(3\ 5\ 15\ 9\ 7)(14),\\ \sigma_2 &= (0\ 10\ 3)(1\ 6\ 7\ 13\ 2\ 9)(4\ 12\ 5\ 11\ 14\ 8\ 15),\\ \sigma_3 &= (0\ 3\ 6\ 13\ 7\ 8\ 9\ 4\ 10\ 5\ 1\ 15\ 14\ 2)(11)(12).\end{aligned}$$

- Permutaciones de la caja de sustitución S_5:

$$\begin{aligned}\sigma_0 &= (0\ 2\ 4\ 7\ 6\ 11\ 15\ 9\ 5\ 10\ 3\ 1\ 12\ 13)(8)(14),\\ \sigma_1 &= (0\ 14\ 8\ 5\ 7\ 1\ 11\ 10\ 15\ 6\ 13\ 9)(2)(3\ 12)(4),\\ \sigma_2 &= (0\ 4\ 10\ 12\ 6\ 7\ 8\ 15\ 14)(1\ 2)(3\ 11\ 5\ 13)(9),\\ \sigma_3 &= (0\ 11\ 9\ 15\ 3\ 7\ 13\ 4\ 1\ 8\ 6\ 2\ 12\ 10)(5\ 14).\end{aligned}$$

- Permutaciones de la caja de sustitución S_6:

$$\begin{aligned}\sigma_0 &= (0\ 12\ 14\ 5\ 2\ 10\ 3\ 15\ 11\ 4\ 9\ 13\ 7\ 8)(1)(6),\\ \sigma_1 &= (0\ 10\ 13\ 11\ 14\ 3\ 2\ 4\ 7\ 5\ 12)(1\ 15\ 8\ 6\ 9),\\ \sigma_2 &= (0\ 9)(1\ 14\ 11\ 10\ 4\ 2\ 15\ 6\ 12)(3\ 5\ 8\ 7)(13),\\ \sigma_3 &= (0\ 4\ 9\ 14\ 8\ 11\ 7\ 10\ 1\ 3\ 12\ 6\ 15\ 13)(2)(5).\end{aligned}$$

- Permutaciones de la caja de sustitución S_7:

$$\begin{aligned}\sigma_0 &= (0\ 4\ 15\ 1\ 11\ 7\ 13\ 10\ 9\ 12\ 5)(2)(3\ 14\ 6\ 8),\\ \sigma_1 &= (0\ 13\ 15\ 6\ 1)(2\ 11\ 12)(3\ 7\ 10\ 5\ 9)(4)(8\ 14),\\ \sigma_2 &= (0\ 1\ 4\ 12)(2\ 11\ 8\ 10\ 6\ 7\ 14\ 9\ 15)(3\ 13\ 5),\\ \sigma_3 &= (0\ 6\ 10)(1\ 11\ 15\ 12\ 14\ 3\ 8\ 9\ 5\ 4)(2\ 13)(7).\end{aligned}$$

- Permutaciones de la caja de sustitución S_8:

$$\begin{aligned}\sigma_0 &= (0\ 13)(1\ 2\ 8\ 10\ 3\ 4\ 6\ 11\ 14\ 12\ 5\ 15\ 7)(9),\\ \sigma_1 &= (0\ 1\ 15\ 2\ 13\ 14\ 9\ 5\ 3\ 8\ 12)(4\ 10\ 6\ 7)(11),\\ \sigma_2 &= (0\ 7\ 2\ 4\ 9\ 6\ 14\ 5\ 12\ 15\ 8)(1\ 11\ 13\ 3)(10),\\ \sigma_3 &= (0\ 2\ 14\ 6\ 8\ 15\ 11)(1)(3\ 7\ 13\ 5\ 10\ 9\ 12)(4).\end{aligned}$$

Ya habíamos comentado arriba, y esto puede entreverse del procedimiento general de operación de las S-cajas, que en las cajas de sustitución recae la parte esencial de la seguridad del cifrado. Más aún, la elección específica de estas S-cajas permite que el cifrado sea resistente a un ataque criptográfico conocido como *criptoanálsis diferencial*. A grandes rasgos, el criptoanálisis diferencial busca romper la seguridad de un cifrado a partir de estudiar cómo pequeños cambios en el mensaje original se traducen al mensaje cifrado. En la época en que

el DES fue presentado al público, el criptoanálisis diferencial no había sido descubierto por la comunidad científica. Sin embargo, en el año de 1990, que fue cuando el criptoanálisis diferencial fue descubierto, el equipo de criptógrafos de IBM declaró que *ellos ya conocían* de ese tipo de ataque, que no lo habían revelado a la comunidad por considerarlo un tipo de ataque bastante avanzado y que *las S-cajas del DES fueron específicamente diseñadas para resistir al criptoanálisis diferencial* [3]. Sobre el criterio de diseño de las S-cajas, lo único que se sabe a ciencia cierta es que fueron diseñadas para satisfacer las siguientes características [12, Subsección 3.3.2]:

1. Cada S-caja recibe bloques de 6 bits y devuelve bloques de 4 bits.
2. Ninguno de los bits resultantes debe ser cercano a una combinación lineal de los bits ingresados.
3. Si el primer y el último bits a ingresar son fijos, y variamos los 4 bits de enmedio, todos los posibles resultados de 4 bits deben poder obtenerse. En otras palabras, las S-cajas consisten efectivamente de permutaciones.
4. Si dos entradas difieren **en un solo bit**, los bloques resultantes difieren **en al menos dos bits**.
5. Si dos entradas difieren **en los dos bits centrales**, sus resultados difieren **en al menos dos bits**.
6. Si dos entradas difieren en los primeros bits y son idénticos en los últimos dos, ambos resultados son diferentes.
7. Dada una diferencia no nula en 6 bits entre entradas, a lo más 8 de las 32 parejas de bits exhibiendo dicha diferencia pueden dar como resultado la misma diferencia.
8. Una **colisión** (diferencia cero en la salida) sólo es posible para tres S-cajas adyacentes.

La permutación P. Después de que las ocho S-cajas producen cada una un bloque de 4 bits, se le aplica una permutación P al bloque resultante de 32 bits. El objetivo de dicha permutación es brindarle difusión al cifrado, ya que ésta hace que los bits que resultan de una S-caja particular en una ronda concreta, en la siguiente ronda entren como argumento a S-cajas diferentes. Más aún, esto produce un *efecto avalancha*, ya que **a partir de la quinta ronda, cada bit resultante depende de todos los bits iniciales** [12, Subsección 3.3.2]. Esta permutación de los bloques de 32 bits, descrita en producto de ciclos, es

$$P = (1\ 16\ 10\ 15\ 31\ 4\ 21\ 32\ 25\ 19\ 24\ 9)(2\ 7\ 28\ 6\ 12\ 26\ 13\ 5\ 29\ 22\ 27\ 30\ 11\ 23\ 3\ 20\ 14\ 18\ 8\ 17).$$

4.4. Esquema de generación de las subclaves

Para terminar de entender cómo opera el cifrado DES, resta explicar cómo se generan las subclaves $k_1, k_2, \ldots, k_{16}$ que entran como argumento a la función f en la ronda correspondiente. Esencialmente, cada una de ellas es una permutación (de algunos) de los bits de la clave original k.

Formalmente, la clave original k del cifrado DES consiste de 64 bits, por lo que en principio existen 2^{64} claves diferentes. Sin embargo, el primer paso del esquema de generación de las subclaves *ignora los bits que se ubican en las posiciones múltiplo de ocho*. De esta manera, solamente 56 de los 64 bits originales intervienen en el cifrado. Por ello, podemos decir con toda justicia que *el tamaño del espacio de claves del DES es de solamente* 2^{56} *bits*.

Elección permutada 1. Una descripción más precisa de cómo pasamos de 64 a 56 bits es diciendo que se aplica una *elección permutada* a los 64 bits. Esta primera elección permutada puede describirse por la tabla de la figura 9, es decir: si $k = k_1 k_2 \ldots k_{64}$ es la clave original de 64 bits, entonces el resultado de $PC_1(k)$ es el bloque de 56 bits $l_1 l_2 \ldots l_{56}$ dado por

$$l_n := k_{F(n)},$$

donde $F(n)$ es el elemento en la entrada n de dicha tabla ($F(1) = 57, F(2) = 49, \ldots$). La función $F : \{1, \ldots, 56\} \to \{1, \ldots, 64\}$ puede describirse analíticamente por $F(n) :=$

$PC-1$							
57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	47	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

Figura 9: La primera elección permutada del esquema de generación de subclaves del DES. Tomada de [12, Tabla 3.13].

$f(g(h(n)))$, donde

$$f(n) := 57n \mod 65, \qquad g(n) := \begin{cases} n+20 & \text{si } 29 \leq n \leq 32, \\ n-20 & \text{si } 49 \leq n \leq 52, \\ n+8 & \text{si } 33 \leq n \leq 36, \\ n-8 & \text{si } 41 \leq n \leq 44, \\ n+16 & \text{si } 37 \leq n \leq 40, \\ n-16 & \text{si } 53 \leq n \leq 56, \end{cases}$$

$$h(n) := n - (n-1 \mod 8) + ((n \mod 8 + 3) \mod 8) \qquad \text{si } n \geq 33.$$

Construcción recursiva de las subclaves. Una vez que de la clave k se ha extraído el bloque $PC_1(k)$ de 56 bits, éste se divide en dos mitades C_0 y D_0 de 28 bits cada una. A partir de ellos, se van construyendo los bloques $C_1, D_1, C_2, D_2, \ldots, C_{16}, D_{16}$ para después extraer de C_i y D_i la subclave k_i.

El procedimiento de construcción de C_i, D_i y la clave k_i a partir de C_{i-1} y D_{i-1} viene descrito en el diagrama de la figura 10. Cada una de las mitades C_{i-1} y D_{i-1} es permutada cíclicamente uno o dos lugares hacia la izquierda (*left shifting* LS_i), según la ronda de la que se trate. En las rondas $i = 1, 2, 9$ y 16, el left shifting LS_i consiste de permutar hacia la izquierda en un lugar, mientras que en el resto de las rondas, LS_i permuta dos lugares hacia la izquierda. El resultado de dichas permutaciones son los nuevos bloques C_i y D_i:

$$C_i := LS_i(C_{i-1}), \qquad D_i := LS_i(D_{i-1}).$$

La clave k_i de 48 bits es extraída del bloque de 56 bits formado por las dos mitades C_i y D_i. De entre todos los 56 bits, se toman 48 de acuerdo a una *segunda elección permutada* PC_2, descrita en la tabla de la figura 11,

$$k_i := PC_2(C_i, D_i).$$

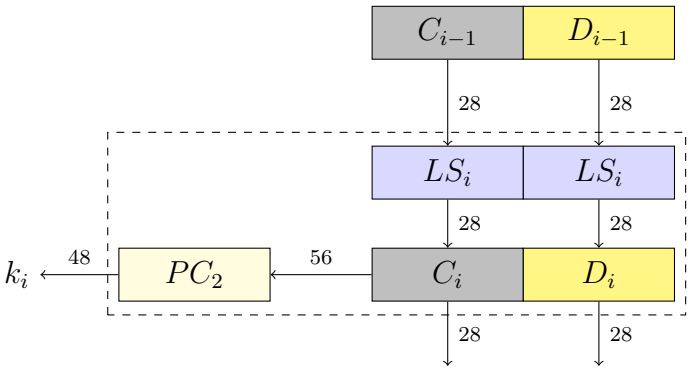

Figura 10: Descripción esquemática de la construcción de la clave k_i de la ronda i.

$PC-2$							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figura 11: La segunda elección permutada del esquema de generación de subclaves del DES. Tomada de [12, Tabla 3.14].

4.5. Descifrando con el DES

Ahora procederemos a explicar cómo es el proceso de descifrado con el *DES*. Un aspecto soprendente de este cifrado es que *las operaciones para cifrar y descifrar son exactamente las mismas*, cambiando únicamente (y sólo un poco) el esquema de generación de claves. Para poder ver esta propiedad, presentamos primero un resumen del cifrado DES:

1. Al bloque original x, de 56 bits, se le aplica la permutación inicial IP (ver subsección 4.2).

2. El resultado de aplicar la permutación se divide en dos bloques L_0 y R_0 de 28 bits cada uno: $(L_0, R_0) := IP(x)$.

3. Para cada $i = 1, 2, \ldots, 16$, se definen $L_i := R_{i-1}$ y $R_i := L_{i-1} \oplus f(R_{i-1}, k_i)$, donde f es la función descrita en la subsección 4.3 y k_i es la subclave de la ronda i.

4. Las mitades L_{16} y R_{16} se intercambian y al bloque resultante se le aplica la inversa de la permutación inicial (ver subsección 4.2),
$$y = IP^{-1}(R_{16}, L_{16}).$$

5. El resultado es el mensaje cifrado $y = \mathrm{DES}_k(x)$.

Ahora bien, el hecho de que el proceso de cifrado y descifrado sean el mismo, salvo por el esquema de generación de las subclaves, se debe a que el DES está basado en una red de Feistel. Vamos a mostrar que efectivamente, podemos recuperar el mensaje original x a partir del mensaje cifrado y aplicando el mismo procedimiento.

Consideremos el mensaje cifrado $y = \mathrm{DES}_k(x)$. Apliquémosle la permutación inicial $IP(y)$ y al resultado dividámoslo en dos bloques izquierdo y derecho L_0^d y R_0^d (el superíndice d indica que estamos en modo de *descifrado*). Luego, para cada $i = 1, 2, \ldots, 16$ defínanse

$$L_i^d := R_{i-1}^d \qquad \text{y} \qquad R_i^d := L_{i-1}^d \oplus f(R_{i-1}^d, k_{17-i}).$$

Escencialmente estamos definiendo las mitades L_i^d y R_i^d tal como lo hicimos en el modo de cifrado, sólo que las subclaves se usan en el orden contrario: la primera subclave es k_{16}, la segunda es k_{15}, etcétera. Finalmente, sea $\widetilde{x} := IP^{-1}(R_{16}^d, L_{16}^d)$. Nuestro objetivo es probar que $\widetilde{x} = x$, es decir, que después de aplicarle al mensaje cifrado $y = \mathrm{DES}_k(x)$ el mismo procedimiento, pero usando las subclaves en el orden contrario, recuperamos el mensaje original x.

Para empezar, vamos a probar por inducción sobre el número de rondas que $L_i^d = R_{16-i}$ y que $R_i^d = L_{16-i}$ para todo $i = 0, 1, 2, \ldots, 16$. Para $i = 0$, tenemos que

$$(L_0^d, R_0^d) = IP(y) = IP(IP^{-1}(R_{16}, L_{16})) = (R_{16}, L_{16}).$$

Supongamos que para $i = 0, \ldots, j$ se cumple que $L_i^d = R_{16-i}$ y que $R_i^d = L_{16-i}$. Para $i = j+1$, tenemos por definición de R y L y por hipótesis de inducción que

$$L_i^d = L_{j+1}^d = R_j^d = L_{16-j} = R_{16-(j+1)} = R_{16-i}.$$

La primera y quinta igualdades son por $i = j + 1$, la segunda y cuarta igualdades son por la definición recursiva de L y la tercera igualdad es hipótesis de inducción. Similarmente,

$$\begin{aligned} R_i^d &= L_{i-1}^d \oplus f(R_{i-1}^d, k_{17-i}) = L_j^d \oplus f(R_j^d, k_{16-j}) \\ &= R_{16-j} \oplus f(L_{16-j}, k_{16-j}) = R_{16-j} \oplus f(R_{16-(j+1)}, k_{16-j}) \\ &= (L_{16-(j+1)} \oplus f(R_{16-(j+1)}, k_{16-j})) \oplus f(R_{16-(j+1)}, k_{16-j}) \\ &= L_{16-i} \oplus (f(R_{16-(j+1)}, k_{16-j}) \oplus f(R_{16-(j+1)}, k_{16-j})) \\ &= L_{16-i}. \end{aligned}$$

Aquí hemos aplicado definición de R^d, $i = j + 1$, hipótesis de inducción, definición de L, definición de R, asociatividad, $i = j + 1$ y el hecho de que $a \oplus a = 0$ si a es un bloque de bits, pues trabajamos módulo 2.

En particular, tenemos que $L_{16}^d = R_0$ y $R_{16}^d = L_0$. Por lo tanto, tomando en cuenta la definición de L_0 y R_0, tenemos

$$\widetilde{x} = IP^{-1}(R_{16}^d, L_{16}^d) = IP^{-1}(L_0, R_0) = IP^{-1}(IP(x)) = x.$$

Esto demuestra que **el descifrado en el DES es el mismo procedimiento que el cifrado, usando las subclaves en orden contrario**.

Subclaves en modo de descifrado. En principio uno pudiera suponer que para aplicar el proceso de descifrado en el DES es necesario generar primero todas las subclaves $k_1, k_2, \ldots, k_{16}$ según se describió en la subsección 4.4. Sin embargo, dicho esquema de generación de las subclaves *permite generar las subclaves en el orden inverso*, es decir, generar primero k_{16}, después k_{15}, etc.

Recordemos que las subclaves de cada ronda se generaban de la siguiente manera:

1. Aplicar la primera elección permutada a la clave k.
2. El resultado se divide en dos bloques de 28 bits cada uno: $(C_0, D_0) := PC_1(k)$.
3. Para cada $i = 1, 2, \ldots, 16$, hacer

$$C_i := LS_i(C_{i-1}), \qquad D_i := LS_i(D_{i-1}), \qquad k_i := PC_2(C_i, D_i),$$

donde LS_i es un *left shifting* sencillo cuando $i = 1, 2, 9$ y 16, y doble en los demás casos y PC_2 es la segunda elección permutada.

Puesto que $k_i := PC_2(C_i, D_i)$, el problema de construir las subclaves en el orden inverso se reduce a obtener C_i y D_i en el orden inverso también. Más aún, puesto que los bloques C_i y D_i se obtuvieron a partir de los anteriores haciendo corrimientos sencillos y dobles hacia la izquierda, según la ronda, podemos generar dichos bloques en el orden inverso *haciendo corrimientos sencillos y dobles hacia la derecha*, siempre y cuando generemos primero los bloques C_{16} y D_{16}.

Vamos a mostrar la siguiente afirmación, sencilla, pero a su vez poco evidente: **los bloques C_{16} y D_{16} son iguales a C_0 y D_0, respectivamente**. En efecto, recordemos que C_{16} se obtiene de aplicar LS_{16} a C_{15}, que a su vez se obtuvo de aplicar LS_{15} a C_{14}, etcétera. Por tanto, C_{16} se obtiene de aplicar $LS_{16}, LS_{15}, \ldots, LS_2, LS_1$ a C_0. Dado que LS_1, LS_2, LS_9 y LS_{16} son corrimientos sencillos y los demás son dobles, en total tenemos 4 corrimientos sencillos y 12 corrimientos dobles, dando un total de 28 corrimientos a la izquierda. En otras palabras, C_{16} se obtiene de C_0 haciendo 28 corrimientos a la izquierda. Recordando que C_0 es un bloque de 28 bits, el aplicarle 28 corrimientos a la izquierda lo deja igual. Por tanto $C_{16} = C_0$. De la misma manera, $D_{16} = D_0$.

Resumiendo la discusión anterior, para generar las subclaves en el orden inverso se procede de la siguiente manera:

1. Aplicar la primera elección permutada a la clave k.
2. El resultado se divide en dos bloques de 28 bits cada uno: $(C_{16}, D_{16}) := PC_1(k)$.
3. Para cada $j = 1, 2, \ldots, 16$, hacer

$$C_{16-j} := RS_j(C_{17-j}), \qquad D_{16-j} := RS_j(D_{17-j}), \qquad k_{17-j} := PC_2(C_{17-j}, D_{17-j}),$$

   donde RS_j es un *right shifting* sencillo cuando $j = 1, 2, 9$ y 16, y doble en los demás casos y PC_2 es la segunda elección permutada.

De esta manera, se generan las subclaves en el orden contrario, y en dicho orden es que se utilizan para descifrar en el DES.

4.6. Ataques al cifrado y alternativas

Para la época en que el DES se utilizó de manera estandarizada, es decir, de 1977 a 1999, el cifrado DES era suficientemente resistente a ataques por fuerza bruta. Esto se debe a que el espacio de claves del DES es de 2^{56}, que para la tecnología de esa época era bastante grande. A lo largo de la década de los 90 se propusieron diferentes ataques al DES con máquinas costosas y especialmente diseñadas para ello, logrando romper el cifrado en cuestión de horas [12, Sección 3.5]. Posteriormente, en el año de 2006, las universidades de Bochum y de Kiel construyeron la máquina llamada COPACOBANA (Cost-Optimized Parallel Code-Breaker), que tuvo un costo aproximado de $10,000 dólares. Dicha máquina en promedio es capaz de descubrir la clave usada en un cifrado DES en solamente 7 días en promedio. Por supuesto, quienes tengan suficientes recursos para ello, digamos gobiernos y grandes empresas, son capaces de invertir recursos para romper el cifrado con ataques por fuerza bruta.

Por otra parte, los ataques analíticos pueden ayudar a romper un cifrado. En 1990 el *criptoanálisis diferencial* fue dado a conocer a la comunidad científica, y tres años después el *criptoanálisis lineal*. Ambos métodos permiten descubrir una clave utilizada en cualquier cifrado de bloque siempre que se conozcan una cierta cantidad de parejas (x, y), donde y es el mensaje cifrado de x. Para el DES, el criptoanálisis diferencial permite descubrir la clave si se conocen 2^{47} parejas, mientras que el criptnoanálisis lineal baja este número a 2^{43}. Estos

números, aunque siguen siendo bastante grandes, son significativamente más pequeños que el espacio de clave del DES, que, aunado al desarrollo tecnólogico dado desde principios de la década de los 90, cada vez era más viable romper el cifrado DES por fuerza bruta o por cualquier otro método.

Hoy en día, un tamaño de clave de 56 bits es considerado pequeño para cuestiones de seguridad, ya que la capacidad de cómputo actual permite romper cifrados de este tamaño de clave mediante un ataque por fuerza bruta relativamente rápido. Por este motivo, se han llegado a utilizar algunas variantes del DES en las que el tamaño de clave es mayor.

El DES triple. Como su nombre lo indica, el DES triple lo que hace es aplicar el DES tres veces, usando tres claves diferentes a la vez. Más precisamente, si κ_1, κ_2 y κ_3 son claves de 56 bits, el triple DES (denotado 3DES o TDEA) aplicado a un mensaje x es

$$y = 3\mathrm{DES}_{\kappa_1,\kappa_2,\kappa_3}(x) := \mathrm{DES}_{\kappa_3}(\mathrm{DES}_{\kappa_2}(\mathrm{DES}_{\kappa_1}(x))).$$

Por supuesto, este procedimiento es aproximadamente tres veces más lento que el DES simple. Sin embargo, esto permite que el cifrado sea mucho más resistente a ataques por fuerza bruta, pues el tamaño de clave se triplica y el espacio de clave consiste de 2^{168} elementos. Otra variante similar a la anterior es

$$y = 3\mathrm{DES}_{\kappa_1,\kappa_2,\kappa_3}(x) := \mathrm{DES}_{\kappa_3}(\mathrm{DES}^{-1}_{\kappa_2}(\mathrm{DES}_{\kappa_1}(x))),$$

donde en el segundo paso se aplica el proceso del DES inverso, es decir, en modo de descifrado. Esta variante tiene la cualidad de que si $\kappa_1 = \kappa_2 = \kappa_3$, el procedimiento coincide con el cifrado DES simple, lo cual es requerido en ciertas aplicaciones. El 3DES es eficiente en hardware pero no en software, y ha sido muy popular en aplicaciones financieras y protección biométrica de información en pasaportes electrónicos.

El DES con blanqueamiento de clave. Otra variante del DES triple utiliza la técnica de *blanqueamiento de clave*. Es un procedimiento muy sencillo en el que se elige una clave k para el DES y dos claves adicionales κ_1, κ_2. Entonces, se utiliza el siguiente esquema de cifrado,

$$y = \mathrm{DES}_{k,\kappa_1,\kappa_2}(x) = \mathrm{DES}_k(x \oplus \kappa_1) \oplus \kappa_2$$

que prácticamente posee la misma velocidad que el DES pero que aumenta significativamente la seguridad del cifrado.

Hay un aspecto criptoanalítico del DES que puede incluso llegar a comprometer la seguridad del 3DES y es la existencia de *claves débiles*, *claves semi-débiles* y *claves posiblemente semi-débiles*. Recordemos que tanto en el proceso de cifrado como el de descifrado del DES se llevan a cabo dieciséis rondas, en las cuales se utiliza una subclave k_i generada a partir de la elección permutada $PC_1(k)$ de una clave original k de 64 bits. En ese sentido, una clave k se dice ser *débil* si resulta que las dieciséis subclaves de rondas $k_1, \ldots, k_{16}$ son todas iguales entre sí. Recordando que las subclaves se obtenen al permutar cíclicamente las mitades de $PC_1(k)$ en una cierta cantidad de lugares según la ronda, la igualdad de todas las subclaves

ocurre cuando los bits de cada mitad de $PC_1(k)$ son todos iguales, es decir, en cada mitad todos 0 o todos 1. Así, las siguientes cuatro son las claves débiles para el DES, escritas en formato hexadecimal:[2]

0101010101010101	FEFEFEFEFEFEFEFE
E0E0E0E0F1F1F1F1	1F1F1F1F0E0E0E0E

Por otro lado, como el 3DES es una composición del DES consigo mismo tres veces, es necesario tomar en cuenta que hay ciertas elecciones de claves κ_1 y κ_2 para las cuales DES_{κ_1} y DES_{κ_2} cifran igual. Esto provoca que en el cifrado del 3DES

$$y = 3\mathrm{DES}_{\kappa_1,\kappa_2,\kappa_3}(x) = \mathrm{DES}_{\kappa_3}(\mathrm{DES}^{-1}_{\kappa_2}(\mathrm{DES}_{\kappa_1}(x)))$$

la operación $\mathrm{DES}^{-1}_{\kappa_2}$ revierte lo hecho por DES_{κ_1}. Luego, ese tipo de par de claves, llamadas *semi-débiles*, debe de evitarse en el 3DES debido a que su uso reduce su nivel de seguridad al del DES. Dichos pares de claves semi-débiles son

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

Además, para el DES también hay 48 claves *posiblemente débiles*, que pueden consultarse en [1, p. 12], con la cualidad de que, de las dieciséis subclaves $k_1, \ldots, k_{16}$, sólo hay cuatro distintas entre sí.

Finalmente, es importante señalar que los cifrados de bloques de 64 bits, como el DES y sus variantes, tienen otro problema importante de seguridad: la alta frecuencia de *colisiones*. Una colisión es simplemente que se obtengan dos textos cifrados iguales, lo cual, cuando se encuentra, permite obtener información significativa acerca del texto original. En el caso de los cifrados de bloque de 64 bits, la probabilidad de obtener una colisión después de cifrar $2^{32} = 4294967296$ bloques es muy alta. Pudiera parecer que son muchos bloques de bits, pero tómese en cuenta que esta cantidad equivale a medio gigabyte de información. Para evitar estos problemas de seguridad, en 2017 el US National Institute of Standards and Technology (NIST) recomendó no utilizar una misma clave en el 3DES para cifrar más de 2^{20} bloques de 64 bits [1, Subsección 3.4] y desde 2024 ha desautorizado su uso en nuevas aplicaciones [2, Sección 2].

[2] En el formato hexadecimal, cada símbolo $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}$ y F representa un número entre 0 y 15, los cuales en binario se expresan con cuatro dígitos 0 o 1. Particularmente, se tiene que $0 = 0000$, $1 = 0001$, $\mathrm{E} = 1110$ y $\mathrm{F} = 1111$. De esta manera, las claves del DES, que constan de 64 bits, se expresan más brevemente con 16 símbolos hexadecimales.

5. Comentarios finales

En esta exposición empezamos describiendo algunos cifrados sencillos, que aunque ya no son de utilidad práctica, nos han permitido entender algunas de las propiedades deseables para los cifrados, así como algunas debilidades típicas que se deben evitar. También hemos presentado el cifrado DES, el cual consiste de un algoritmo más complejo cuya estructura es la de una red de Feistel y que posee fuertemente las propiedades de confusión, difusión, no linealidad y resistencia al criptoanálisis diferencial. Además, posee un espacio de claves de tamaño 2^{56}, el cual es bastante grande para la época en que fue dominante y por lo tanto resistente a los ataques por fuerza bruta. El DES fue un cifrado estandarizado y ampliamente utilizado de 1977 a 1999. Originalmente había sido pensado para utilizarse por solamente 10 años, hasta 1987. Sin embargo, como no se le encontraron debilidades serias, se extendió su uso hasta 1999.

Este cifrado nos permitió entender la complejidad que está detrás de los cifrados modernos, los cuales realizan operaciones por computadora que difícilmente podríamos realizar a mano. Sin duda alguna, la tecnología ha permitido avanzar en los aspectos de la seguridad, abriendo la puerta a todas las aplicaciones que actualmente forman parte de nuestra cotidianeidad.

En 1997, el US National Institute of Standards and Technology (NIST) abrió una convocatoria para un nuevo cifrado estandarizado, el *Advanced Encryption Standard (AES)* para reemplazar al DES. Esta fue una convocatoria totalmente abierta en la que el NIST fungió como administrador, a diferencia de con el DES. En cada una de las tres rondas de selección del AES, el NIST y la comunidad científica discutieron las ventajas y desventajas de cada cifrado hasta que se seleccionó un ganador.

Los requisitos que el NIST impuso a las propuestas fueron los siguientes:

1. Que fuera un cifrado de bloques de 128 bits.
2. Que sea capaz de funcionar con claves de longitudes de 128, 192 y 256 bits.
3. Eficiencia en software y hardware.

En agosto de 1999, fueron anunciados cinco finalistas: *Mars*, de la IBM; *RC6* de los Laboratorios RSA; *Rijndael*, diseñado por los criptógrafos belgas Joan Daemen y Vincent Rijmen; *Serpent*, diseñado por Ross Anderson, Eli Biham y Lars Knudsen; y *Twofish*, diseñado por criptógrafos de la Counterpane Internet Security, de Princeton y Berkeley. Después de un año de exhaustivos análisis entre los cifrados presentados, se anunció como ganador al cifrado Rijndael, pasando a ser el nuevo *Advanced Encryption Standard*.

Este ejemplo nos enseña que los cifrados no son eternos y que aunque un cifrado sea capaz de responder a las necesidades de una época concreta, eventualmente tanto la tecnología como los avances de la criptografía hacen que los cifrados deban ser reemplazados por mejores algoritmos. Actualmente, la expectativa es que la computación cuántica se desarrolle de manera significativa y sea criptográficamente relevante en la próxima década. De hecho, desde hace tiempo ya existen los llamados *algoritmos cuánticos* que, en principio, podrían llegar a ser implementados en computadoras cuánticas y ser capaces de romper muchos de los cifrados que hoy son considerados seguros, incluyendo algunas versiones del AES.

Por ejemplo, el *algoritmo de Grover* (1996) es un algoritmo cuántico de búsqueda que permite con una alta probabilidad realizar ataques por fuerza bruta exitosos en un tiempo del orden de la raíz cuadrada del tamaño del espacio de claves [7]. En el caso del AES-128, es decir, la versión del AES con claves de 128 bits, el espacio de claves es de tamaño 2^{128}. En teoría, cuando las computadoras cuánticas lleguen a ser capaces de implementar el algoritmo de Grover, el nivel de seguridad del AES-128 bajaría a $\sqrt{2^{128}} = 2^{64}$, volviéndose vulnerable a ataques por fuerza bruta. Similarmente, el nivel de seguridad del AES-196 bajaría de 2^{196} a 2^{98} mientras que el del AES-256 bajaría a 2^{128}, siendo este último tamaño de clave todavía resistente a ataques por fuerza bruta.

Como puede intuirse, existe una preocupación real de que dentro de pocos años la computación cuántica vuelva inseguros los cifrados más utlizados actualmente. Debido a ello, en 2016 el NIST lanzó convocatorias para la adopción de algoritmos estandarizados que sean resistentes a la computación cuántica. En 2022, se seleccionaron cuatro algoritmos [10], de los cuales tres ya han sido estandarizados en 2023 y se espera que se seleccionen y estandaricen más algoritmos para su uso y estudio. De esta manera, estamos viviendo cómo se inaugura una nueva etapa de la criptografía postcuántica.

Agradecimientos

El autor agradece a los tres revisores anónimos por tomarse el tiempo de revisar y señalar varios errores que había en la primera versión del artículo, así como por sugerir la inclusión de referencias adicionales que, sin duda, han dado mayor realce a los temas aquí discutidos.

Referencias

[1] E. Barker and N. Mouha, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher," *Special Publication (NIST SP) 800-67 Rev. 2.*, pp. 1–25, 2017. DOI: 10.6028/NIST.SP.800-67r2. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-67r2

[2] E. Barker and A. Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths," *Special Publication (NIST SP) 800-131A Rev. 2.*, pp. 1–27, 2019. DOI: 10.6028/NIST.SP.800-131Ar2. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-131Ar2

[3] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243–250, 1994. DOI: 10.1147/rd.383.0243. [Online]. Available: https://doi.org/10.1147/rd.383.0243

[4] J. A. de la Peña, *Álgebra en todas partes.* Fondo de Cultura Económica, 1999.

[5] J. v. z. Gathen, *CryptoSchool*, 1st ed. Springer Berlin, Heidelberg, 2016. [Online]. Available: https://doi.org/10.1007/978-3-662-48425-8

[6] L. C. Grove, *Algebra.* Academic Press, 1983.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing.* New York, NY, USA: Association for Computing Machinery, 1996. DOI: 10.1145/237814.237866 p. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[8] T. Kelly, "The myth of the skytale," *Cryptologia*, vol. 22, no. 3, pp. 244–260, 1998. DOI: 10.1080/0161-119891886902. [Online]. Available: https://doi.org/10.1080/0161-119891886902

[9] G. Morales-Luna, "Sobre el telegrama zimmerman," *CINVESTAV-IPN*, 2016, 21 de diciembre. [Online]. Available: https://delta.cs.cinvestav.mx/~gmorales/

[10] NIST, "Nist announces first four quantum-resistant cryptographic algorithms," *NIST news*, 2022. [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms

[11] I. M. Niven, H. S. Zuckerman, and H. L. Montgomery, *An Introduction to the Theory of Numbers*, 5th ed. Jhon Wiley, 1991.

[12] C. Paar, J. Pelzl, and T. Güneysu, *Understanding Cryptography*, 2nd ed. Springer Berlin, Heidelberg, 2024. [Online]. Available: https://doi.org/10.1007/978-3-662-69007-9

ISSN 2448 5365
Universidad de Sonora

[13] F. Russel, *Information Gathering in Classical Greece*. University of Michigan Press, 1999.

Cómo citar este artículo: E. Velasco-Barrera, "Criptografía de los cifrados de bloque", Sahuarus. Revista Electrónica de Matemática, vol. 8, no. 1, pp. 45 – 82, 2024. https://doi.org/10.36788/sah.v8i1.100