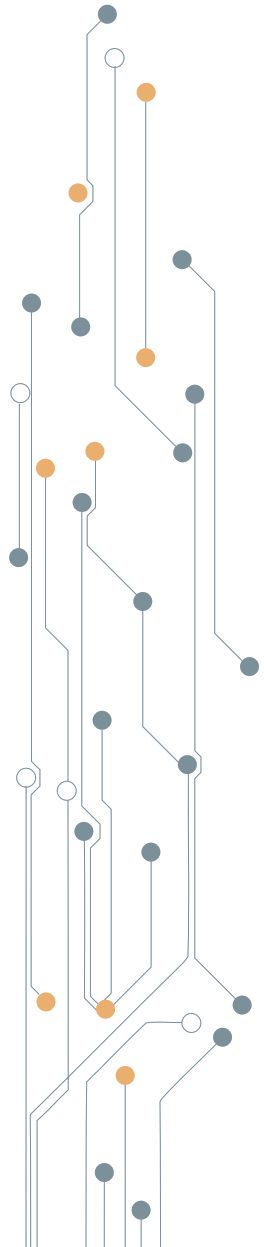
Algoritmos de cifra simétrica en bloque

Telefónica EDUCACIÓN DIGITAL

Índice



1. Esquema y fundamentos de la cifra en bloque

En la cifra simétrica en bloque la información a cifrar o texto en claro se agrupa en bloques n de bits, típicamente 128 bits o 16 octetos. Estos bloques entran al cifrador y se mezclan con la clave mediante diversas operaciones de permutación y sustitución, tras lo cual el algoritmo entrega bloques de salida del mismo tamaño que el bloque de la entrada, pero cifrados.

Para que los bits del texto en claro se mezclen lo suficiente con los bits de la clave, habrá que aplicar permutaciones y sustituciones esos bits varias veces. En criptografía a estas operaciones repetitivas se les llama vueltas o rondas del algoritmo. Con ello se logra, entre otras características interesantes, el denominado efecto de avalancha, esto es, que el cambio de un solo bit en la entrada (o en la clave) produzca un cambio en aproximadamente el 50% de los bits de salida. En otras palabras, que cada bit de salida sea una función compleja que depende en un 50% de cada bit de entrada.

La figura 6.1 muestra cómo se forman los bloques de cifra.

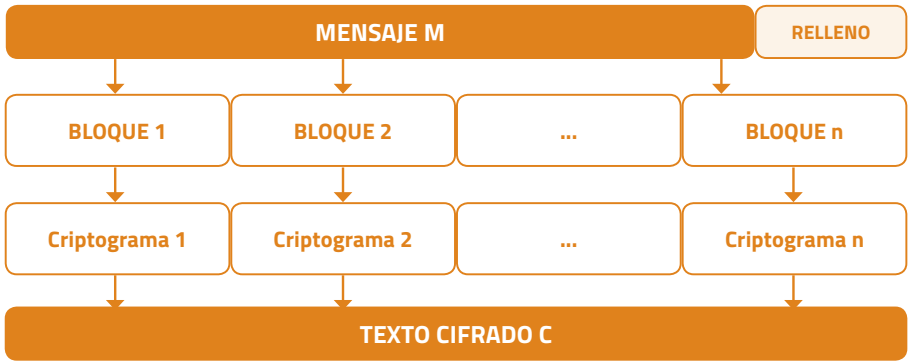


Figura 6.1. Bloques en la cifra simétrica.

De la misma forma que los bloques 1, 2, etc. del texto en claro corresponden a información en claro cuyos bits están concatenados formando el mensaje M, el criptograma final C será el resultado de la concatenación de los criptogramas 1, 2, etc., también llamados subcriptogramas. Observa que se ha añadido al final un relleno por si el texto en claro no es congruente con el tamaño del módulo.

Cada bloque se cifra durante varias vueltas en las que se usa una subclave, que se calcula a partir de la clave de cifra K con una función que se conoce como expansión de clave, de forma que en cada vuelta ésta sea distinta. La figura 6.2 muestra el esquema de un cifrado simétrico y las operaciones características que se realizan en cada fase del algoritmo.

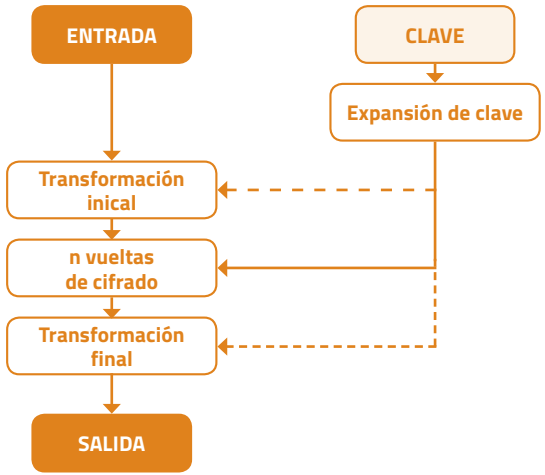


Figura 6.2. Esquema de la cifra simétrica.

Modos de cifra

Aunque existen media docena modos de cifra, los más conocidos son:

1. Modo de cifra libro electrónico de códigos ECB: Electronic Code Book.
2. Modo de cifra por encadenamiento de bloques CBC: Cipher Block Chaining.
3. Modo de cifra contador CTR: Counter.

El primero de ellos **ECB** es el modo de cifra por defecto, como viene escrito el código básico del algoritmo, pero que no debe usarse pues presenta varias vulnerabilidades. El segundo modo CBC, ha sido el más popular de todos y se sigue usando de manera mayoritaria en las comunicaciones seguras en Internet vía SSL/TLS. Por último, el tercero de ellos CTR, presenta interesantes ventajas con respecto al modo CBC y se va abriendo camino como el modo de cifra más recomendado actualmente.

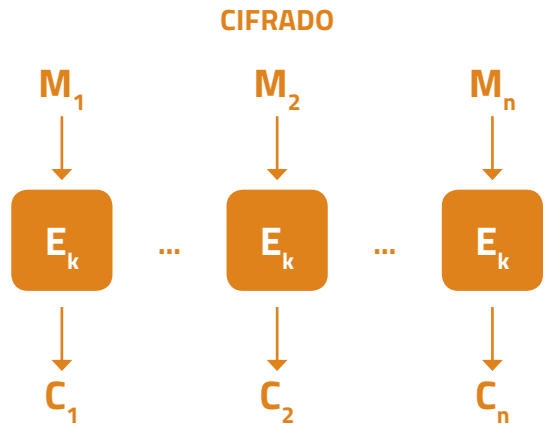


Figura 4.1. Clasificación de los sistemas de cifra moderna.

El modo de cifra por libro electrónico de códigos ECB, Electronic Code Book, consiste en cifrar bloques de texto en claro de manera independiente. Así, cada bloque de x bits del texto en claro se cifra con la clave K del algoritmo y entrega un bloque de texto cifrado de x bits. No se recomienda el uso de este modo porque es vulnerable a ataques por inicios y finales iguales entre dos documentos diferentes, así como a ataques por repetición de bloques elegidos o bien modificados. No obstante, el modo de cifra ECB tiene como aspecto positivo que al cifrarse bloques de texto en claro manera independiente, cualquier error de transmisión afectará solamente al bloque en cuestión y no a todo el criptograma.

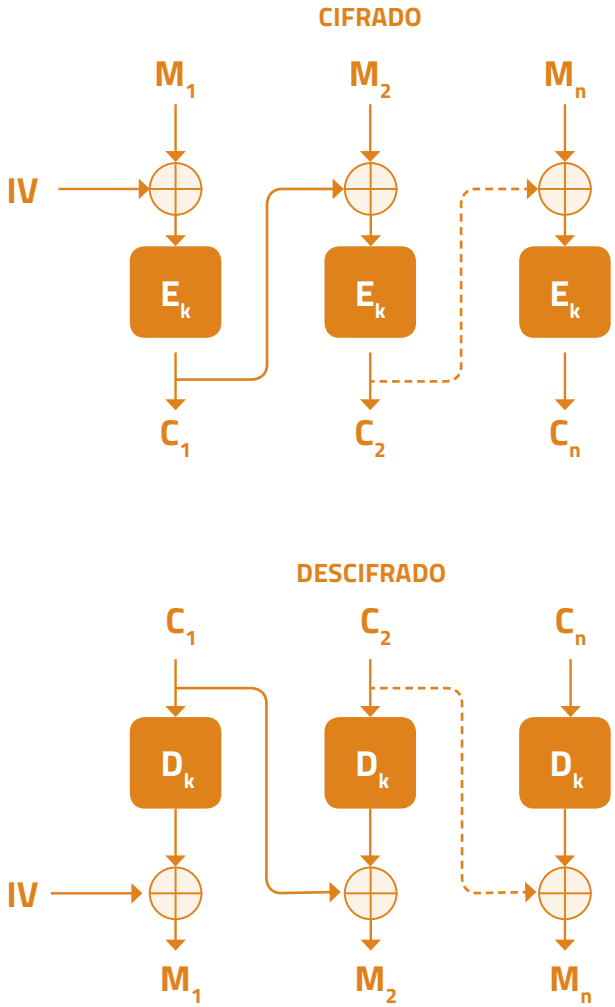


Figura 6.4. Modo de cifra CBC, cifrado y descifrado.

En el modo de cifra por encadenamiento de bloques cifrantes **CBC,** Cipher Block Chaining, se usa una segunda clave de igual tamaño que el bloque de texto a cifrar llamada vector inicial IV, que se recomienda sea también secreta aunque ello no sea obligatorio, con la cual se realiza un OR exclusivo con el primer bloque del texto en claro a cifrar, antes de comenzar la cifra. El resultado de la cifra de esta nueva entrada con la clave K se usará como vector inicial del siguiente bloque de texto a cifrar, y así sucesivamente. Por tanto, se va encadenando el cifrado anterior con el nuevo cifrado usándolo como vector de suma OR exclusivo para el nuevo bloque de texto a cifrar.

Para descifrar con este método, una vez se ha aplicado el algoritmo de descifrado al primer criptograma con la clave K, se recupera el texto en claro realizando una suma OR exclusivo con ese vector inicial IV. Los siguientes bloques de criptograma se descifran de igual manera, pero utilizando ahora como vector inicial el criptograma anterior, que hemos guardado previamente en un archivo temporal.

Observa que ahora en este modo de cifra un error en la transmisión afectará a todo el resto del criptograma. Unido a esto, aparece el problema asociado de que CBC no permite paralelizar la operación de descifrado, algo importante cuando hablamos de cifrado de grandes volúmenes de información, que será solucionado en el tercer modo de cifra a estudiar, el modo contador.

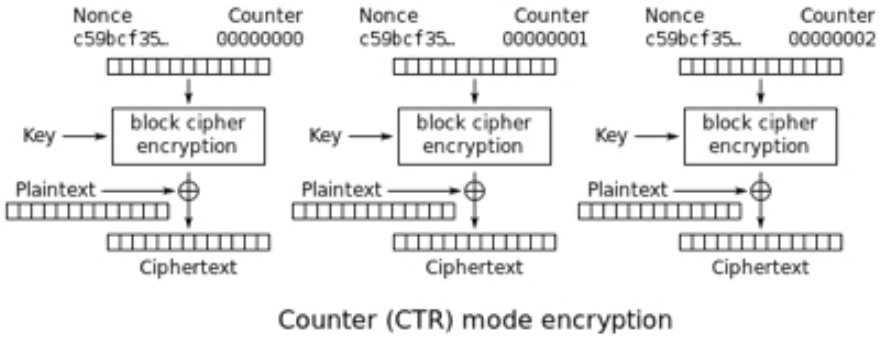


Figura 6.5. Modo de cifra CTR.

Como se observa, en el modo **CTR** el vector nonce parte de un valor inicial para la cifra del primer bloque del texto en claro y para los siguientes bloques su valor se va incrementando en una unidad. De esta manera por una parte se permite la cifra en paralelo de varios bloques a la vez, en tanto ahora el resultado de la cifra de un bloque de texto en claro anterior no afectará a la cifra del bloque siguiente como sucedía con el modo CBC y, además, un error de transmisión afectará solamente al bloque en cuestión y no a todo el mensaje.

Otra característica del modo CTR es que convierte un cifrador de bloque en un cifrador de flujo, en el sentido de que cada bloque se va cifrando con una clave diferente que es el resultado de la cifra del vector nonce que se incrementa en una unidad tras cada bloque de texto en claro. Por lo tanto, la clave de cifra se convierte en una secuencia de clave como sucedía en los cifradores de flujo.

2. El algoritmo DES

El Data Encryption Standard ha sido el estándar mundial de cifra simétrica utilizado durante 25 años, hasta finales de 1999. Aunque el algoritmo nunca ha sido criptoanalizado ni se le han encontrado debilidades graves, tenía fecha de caducidad anunciada debido a que la clave real era sólo de 56 bits y su tamaño fijo. En los años 70 romper por fuerza esos 56 bits era computacionalmente imposible, pero a finales de los años 90 esto podía realizarse mediante un ataque distribuido y romper la clave en menos de un día.

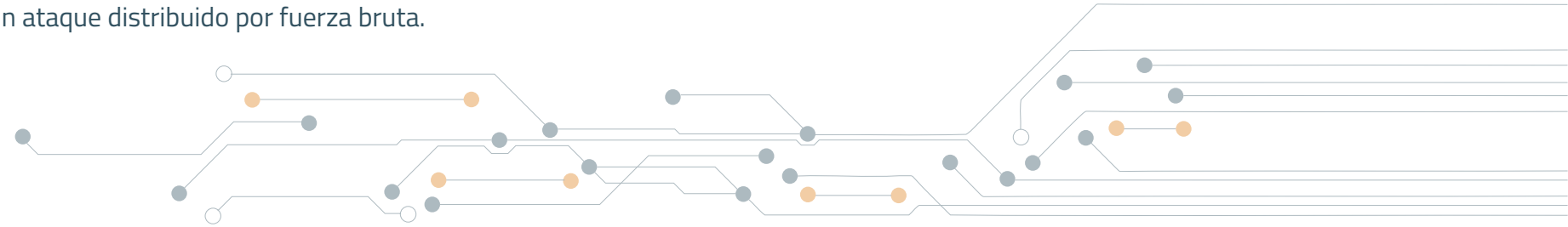
La clave en un sistema de cifra en bloque será un número aleatorio, secreto y único que se intercambian previamente los interlocutores, o bien un número que genera quien desea cifrar un documento o archivo de forma local, lo que se conoce como cifrado convencional.

Se conoce como espacio de claves a todos los valores posibles que ésta podría obtener. En el caso anterior de una clave de 64 bits, estos valores van desde el valor 0, una cadena de 64 ceros, hasta 264 - 1 = 18.446.744.073.709.551.615, una cadena de 64 unos. Como la clave de cifra es un único número dentro de ese espacio de claves, y ésta no cambia dinámicamente como sucede en la cifra en flujo, los algoritmos simétricos deben diseñarse con un tamaño de clave que no haga viable un ataque distribuido por fuerza bruta.

Obviamente, para encontrar un valor cualquiera dentro de un espacio de claves de n bits, hará falta en media realizar 2n-1 intentos o, lo que es lo mismo, recorrer en media la mitad de dicho espacio de claves.

El DES utiliza una clave de 64 bits pero que se ve reducida a 56 bits nada más comenzar a ejecutarse el algoritmo, puesto que los valores ASCII que se usaban en aquellos años, o bien claves de valores numéricos, reservaban el octavo bit para paridad.

El algoritmo DES un cifrador de tipo Feistel. Esto quiere decir que las operaciones de cifra de un bloque de texto en claro se realizan sólo sobre una mitad de ese bloque de texto en claro y no sobre el bloque completo. Como el algoritmo tiene varias vueltas, en la siguiente vuelta se intercambian esas mitades tal y como se muestra en la figura 6.6. Cifrará bloques de texto de 64 bits, con una clave de 64 bits principal que se reduce a 56 bits, realiza 16 vueltas o rondas, donde se intercambian las posiciones de las mitades del bloque de texto en claro como se ha comentado, y en cada vuelta usa una subclave ki de 48 bits, derivada de la clave principal. Para lograr los objetivos de difusión y confusión, en cada vuelta se realizan operaciones de sustitución y de permutación.



La figura 6.6 muestra el bloque de 64 bits de texto en claro que se divide en dos mitades de 32 bits cada una, la mitad izquierda Ai y la mitad derecha Bi. La mitad derecha se mezcla con la clave de esa primera vuelta en una función denominada F, y su resultado se suma or exclusivo con la mitad izquierda del texto en claro. Hecho esto, este resultado se pasa como mitad derecha de texto en claro de la vuelta siguiente y la mitad derecha de la vuelta anterior se pasa como mitad izquierda de la nueva vuelta. Este proceso se repite durante las 16 vueltas del algoritmo, tras lo cual se obtiene un bloque de 64 bits cifrados correspondiente a la cifra de ese primer bloque de texto, lo que se repite hasta el último bloque de texto en claro.

Para poder realizar la operación XOR entre la mitad izquierda del texto en claro y el resultado de varias operaciones hechas sobre la mitad derecha, se procede a ampliar los 32 bits de texto en claro hasta los 48, añadiendo 16 bits, y rebajar los 56 bits de la clave a 48, reduciendo o eliminando 8 bits. Esto se hace así porque la operación más importante de la función F son las cajas S que convierten una cadena de 48 bits en una cadena de 32 bits (los necesarios para el XOR final de cada vuelta) y que, al ser una función no lineal, dota de fortaleza al algoritmo.

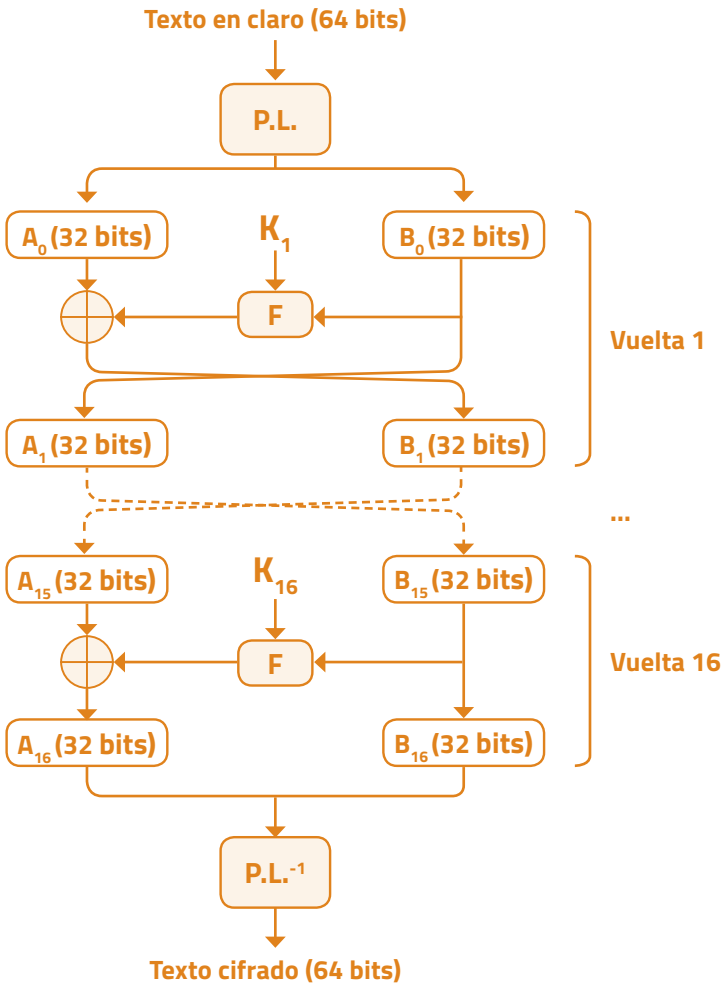


Figura 6.6. Esquema del algoritmo DES.

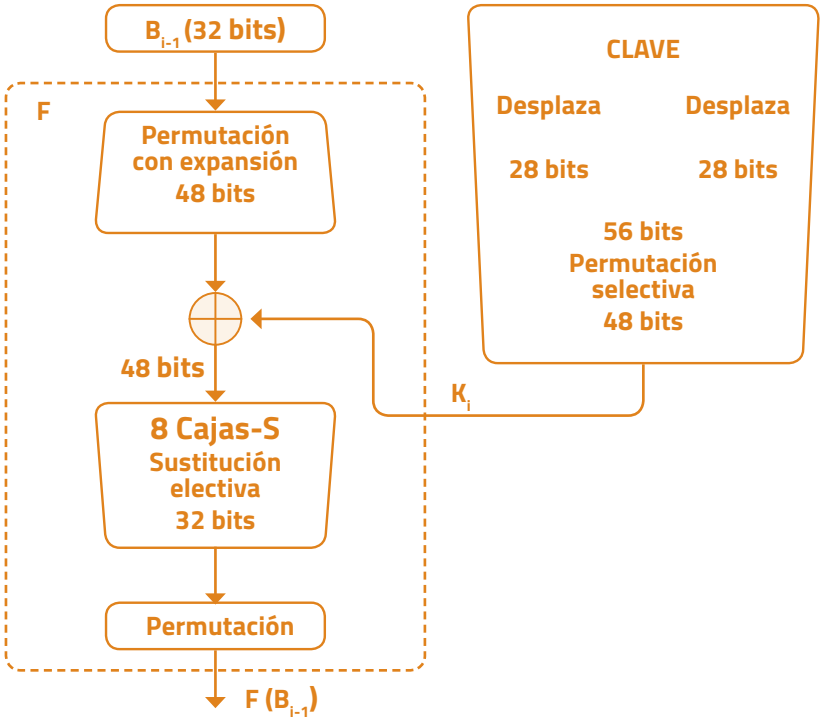


Figura 6.7. Operaciones en una vuelta del DES.

Existirán 8 cajas-S de forma que los 48 bits se distribuirán en ellas. Así, la caja S1 tendrá como entrada los bits 1 al 6, la caja S2 los bits 7 al 12, ... y la caja S8 los bits 41 al 48. En cada caja entran por tanto 6 bits y mediante un proceso que veremos a continuación salen 4 bits.

De los seis bits que entran en cada una de las 8 cajas, se leen los valores de los 2 bits de los extremos, que entregan 4 valores posibles (00, 01, 10 y 11) y con ello tenemos un dato, un número decimal del 0 al 3 que marcará la fila de esa caja. A continuación se leen los cuatro bits interiores de dicha cadena de seis bits, que entregan ahora 16 valores posibles (0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111), es decir números decimales entre 0 y 15, y que nos señalarán la columna de esa caja.

S_1	COLUMNAS															
FILAS		0	1	2	3	4	5	6	7	8	9	10	11	12	12	14	15
	0	14	4	15	1	2	15	11	8	5	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figura 6.8. Caja S_1.

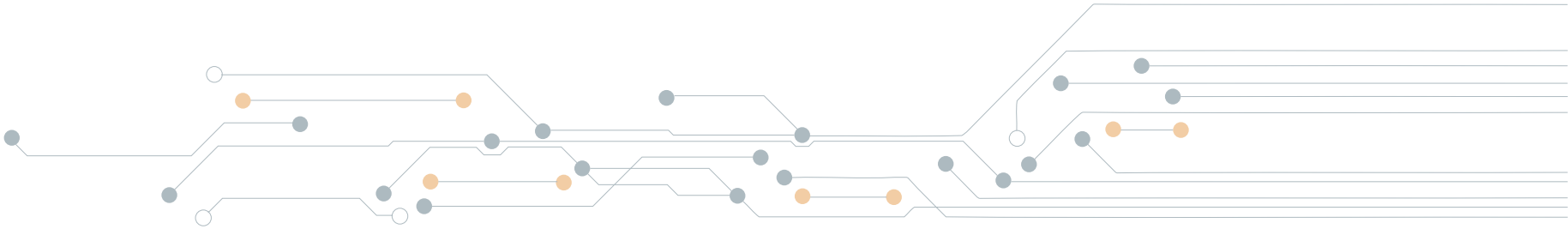
Por ejemplo, si los primeros 6 bits de la cadena de 48 bits, que entran por tanto en la caja S_1, son **010001**, entonces los bits en naranja 01 marcan la fila 1 y los bits en azul **1000** marcan la columna 8. La intersección será el número decimal 10 que en binario es 1010.

Las cajas S dan fortaleza al DES porque romper el algoritmo implica romper dichas cajas. Observa que en el ejemplo de la figura 6.8, la salida decimal 10 puede obtenerse además con estas otras tres entradas: fila 0 columna 9, fila 2 columna 13 y fila 3 columna 12. Como hay 8 cajas S, para romper el algoritmo en sus 16 vueltas habrá que realizar $4^{8^{16}}$ operaciones, es decir 2^{256} intentos, que es mucho más difícil que romper el criptograma por fuerza bruta descifrando con las 2^{56} claves posibles.

Si para cifrar se usaban las 16 subclaves de 48 bits desde K_1 hasta K_{16}, para descifrar se usa el mismo algoritmo pero las claves se utilizan ahora desde K_{16} a K_1. Esto es así porque en cada vuelta se produce un desplazamiento de las dos mitades de 28 bits de la clave de -1 o -2 bits en donde el signo negativo quiere decir que es un desplazamiento hacia la izquierda. Si sumamos el desplazamiento aplicado a cada cadena de 28 bits en las 16 vueltas, observamos que es igual a 28 y por lo tanto las cadenas izquierdas C_0 y C_{16} estarán en fase, lo mismo que las cadenas derechas D_0 y D_{16}.

Como los 56 bits de $C_{16}D_{16}$ están en fase con los 56 bits iniciales C_0D_0, para descifrar se usarán los mismos desplazamientos pero en sentido inverso, es decir +1 y +2, y así encontrar las claves que se usaron en el cifrado, recorriendo por tanto el algoritmo de forma inversa.

El DES ha sufrido diversos ataques distribuidos exitosos en red conocidos como DES Challenge desde finales del siglo pasado, propuestos por RSA Laboratories, básicamente porque su espacio de claves era muy pequeño. Después de 3 desafíos que comienzan en enero de 1997, DES finalmente sucumbe en enero de 1999. Se unen la máquina DES Cracker y distributed.net con 100.000 ordenadores conectados en Internet para romper la clave en solamente 22 horas, evaluando casi 250.000 millones de claves por segundo.



3. El algoritmo 3DES

Como la clave tan pequeña del DES lo hacía vulnerable ya en la década de los 90, en 1998 IBM propone un cifrado múltiple, es decir, se vuelve a cifrar el criptograma una o más veces con otras claves. En estos sistemas no lineales esto hace que la clave efectiva aumente de tamaño, en el sentido de que cada vez se necesitan más operaciones para poder romperla. Nace así el triple DES o 3DES.

No se usa un cifrado doble porque mediante un ataque con texto en claro conocido, denominado Meet In The Middle (no confundir con Man In The Middle), se concluye que la clave del sistema aumenta tan sólo en un bit. Demasiado esfuerzo computacional para tan poco resultado. En el cifrado triple, si se cifra con tres claves, K_1, K_2 y K_3, todas de 56 bits, la fortaleza del sistema de cifra será de 56*3 = 168 bits.

El sistema que se utiliza, propuesto por IBM, es el denominado EDE, Encrypt Decrypt Encrypt, que se muestra en la figura 6.9.

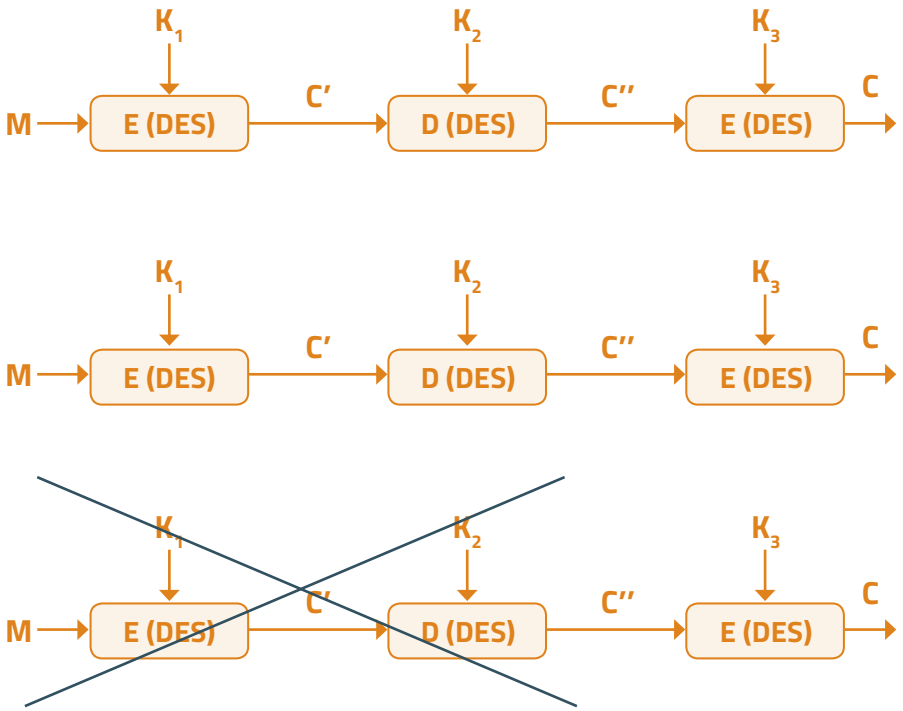
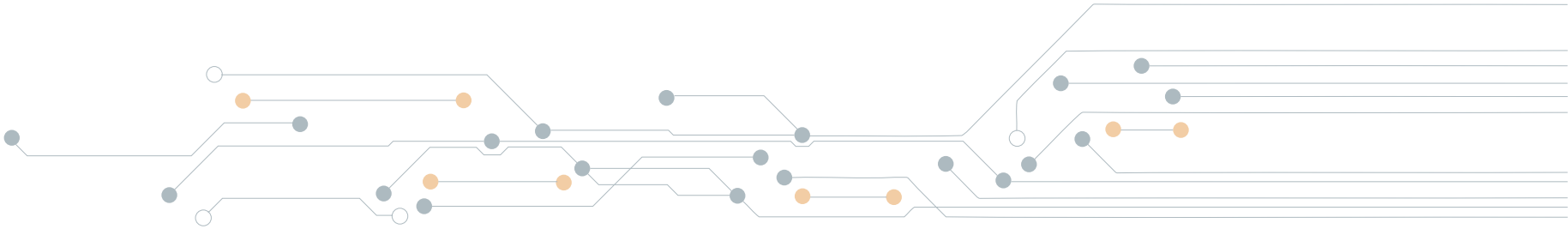


Figura 6.9. Esquema del 3DES en modo EDE.

Observa en el primer esquema de la figura 6.9 que primero se cifra con k_1, a continuación se descifra con k_2 y finalmente se vuelve a cifrar ahora con k_3. Esto sería igual que realizar tres cifrados con tres claves distintas y la fortaleza total será de 256*3 = 2168, un valor muy seguro.

Una segunda opción del algoritmo 3DES es la que se muestra en los dos esquemas siguientes, usar k_1 y k^2 como claves diferentes en el primero y hacer $k^2 = k^1$ en el segundo. En el primero de dos claves la fortaleza del sistema será igual a $2^{56*2} = 2^{112}$. Tiene como ventaja que sólo hay que intercambiar con el destino dos claves, no tres, pero su fortaleza el algo débil. Además permitía la opción que los interlocutores definiesen las tres claves iguales, esto es $k_1 = k_2 = k_3 = k$, lo cual era muy interesante en aquellos primeros años en que se usaba este algoritmo, en tanto hacía compatibles sistemas modernos que tuviesen implementado el cifrado 3DES con sistemas más antiguos que tuviesen implementado solamente el DES.



4. El algoritmo AES

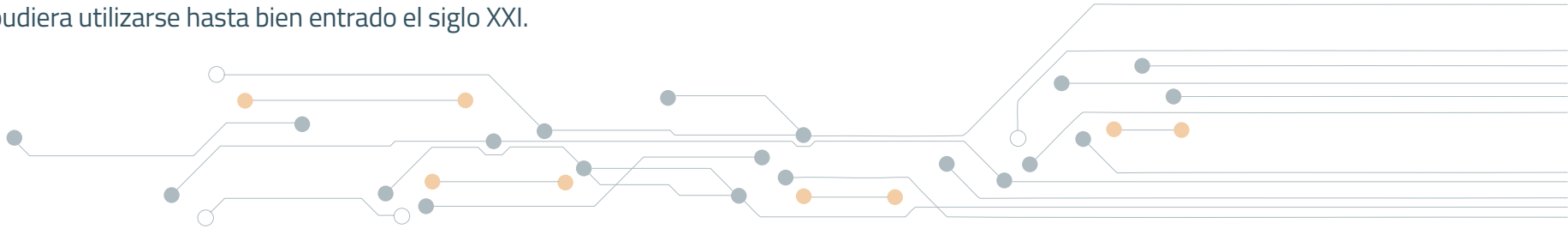
El algoritmo AES nace como el sustituto del DES como estándar mundial de cifra simétrica. Haciendo un poco de historia, el DES se adopta como estándar en 1976, el NIST certifica como nuevamente el estándar primero en 1987 y después en 1993. Durante esos años se populariza como algoritmo de cifra en todo el mundo y su uso más amplio lo encontramos en el cifrado de la información intercambiada en transacciones de dinero entre un cajero automático y el banco respectivo. En 1997 el NIST ya no certifica al DES como estándar y llama a un concurso internacional para buscar un nuevo estándar mundial de cifra que se denominará AES, acrónimo de Advanced Encryption Standard. Precisamente entre 1997 y 1999 el DES se había enfrentado a los cuatro ataques o desafíos que hemos visto, acción que impulsa y promociona la compañía RSA, a la fecha el estándar de cifra asimétrica, para demostrar que la cifra simétrica estándar era muy débil.

En las bases de la convocatoria del NIST se establece que el nuevo estándar debería soportar una longitud de bloque de 128 bits y una longitud de clave de 128, 192 y 256 bits, con la intención de que el nuevo estándar pudiera utilizarse hasta bien entrado el siglo XXI.

Se presentan 15 candidatos al concurso y después de más dos años, a finales del año 2000, se proclama vencedor al algoritmo Rijndael de los investigadores belgas Vincent Rijmen y Joan Daemen. En noviembre de 2001 el NIST anuncia oficialmente a través del Federal Information Processing Standard Publications FIPS 197 que el nuevo estándar para cifrado simétrico del siglo XXI será el AES.

AES es un algoritmo no de tipo Feistel, que procesa por tanto bloques completos de texto en claro de 128 bits, con claves estándar de 128, 192 o 256 bits. Para ello usa una matriz de estado de tamaño 4x4, cuyas 16 celdas o bytes van cambiando de valor de acuerdo a los procesos que ejecuta el algoritmo. En el cifrado se utilizan técnicas de sustitución y permutación, en algunos casos con operaciones polinómicas dentro de un cuerpo. Todas las operaciones de cifra dentro de la matriz de estado se realizan sobre bytes, en palabras de 32 bits que se escriben de arriba hacia abajo y de izquierda a derecha.

La figura 6.10 muestra el esquema del algoritmo AES.



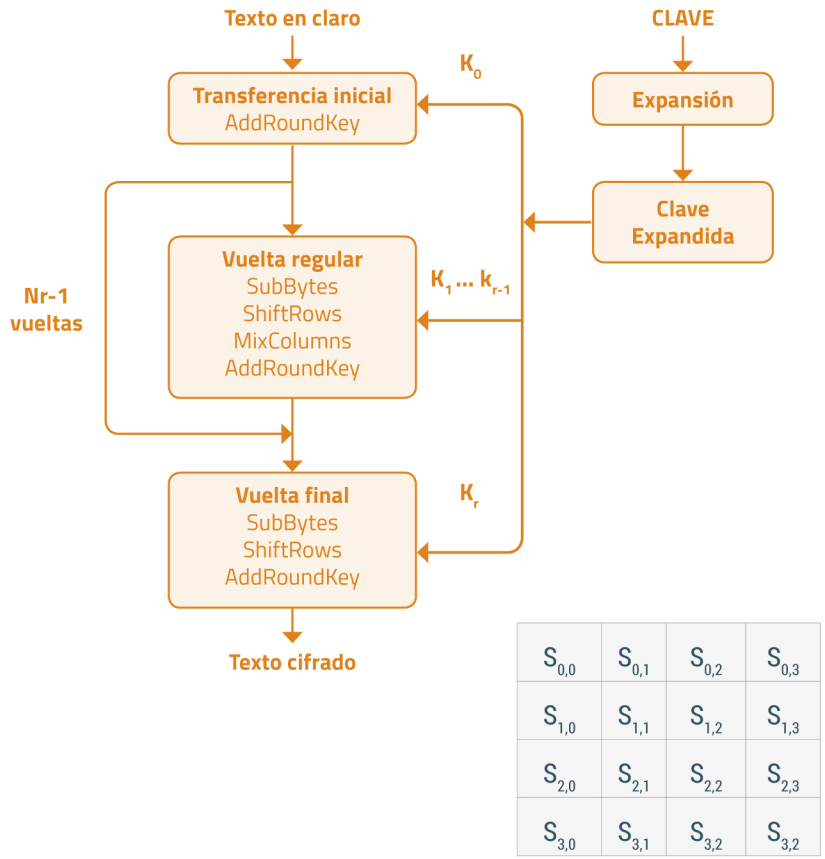
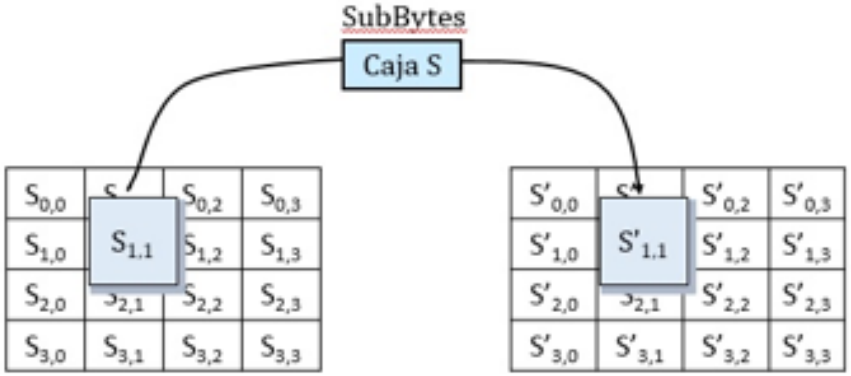


Figura 6.10. Esquema del algoritmo AES y matriz de estado.

Las cuatro operaciones básicas de cifrado en el AES son AddRoundKey, SubBytes, ShiftRows y MixColumns.

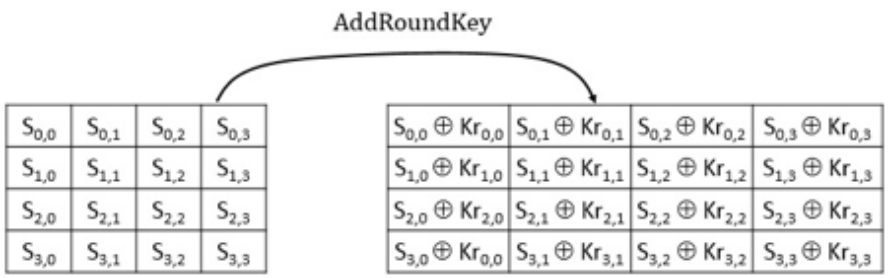
- La operación AddRoundKey ejecuta la suma or exclusivo entre los bytes del mensaje y los bytes de la clave.
- La operación SubBytes ejecuta una sustitución de cada uno de los 16 bytes de la matriz de estado mediante una tabla.
- La operación ShiftRows consiste en una permutación de las filas del estado de forma que la primera fila no rota, la segunda rotará un byte, la tercera rotará 2 bytes y la cuarta rotará 3 bytes.
- La operación MixColumns es algo más compleja y consiste en multiplicar cada una de las columnas de la matriz de estado por un polinomio fijo.

Operación SubBytes



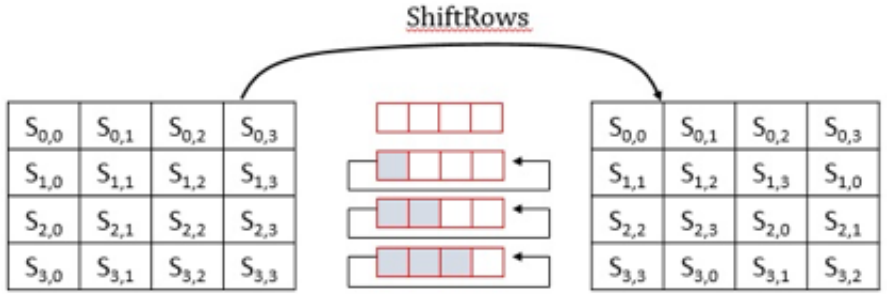
Su finalidad es introducir no linealidad en el proceso. Cada uno de los bytes de la matriz de estado es sustituido a través de una Caja S de 8x8, es decir ante 8 bits de entrada, la salida muestra un valor distinto pero también de 8 bits. Aunque hay una función matemática para encontrar estos valores y cuyo diseño pretende minimizar la relación entre la entrada y la salida, es más fácil observar que se trata de una matriz de tamaño 16x16, es decir 256 celdas en donde se introducen los 256 valores posibles en hexadecimal de 8 bits, desde el 00 hexadecimal igual a 0, hasta la FF hexadecimal igual a 255, una tabla de reemplazo.

Operación AddRoundKey



Se realiza una suma módulo 2 (XOR) del estado con la subclave Kr, que es la última de las subclaves obtenidas a partir de la clave maestra K y siempre tendrá una longitud de 128 bits (16 bytes).

Operación ShiftRows



En esta transformación se permutan cíclicamente los contenidos de las filas del estado. Tiene por objeto aumentar la difusión. Concretamente, la fila 0 no desplaza, la fila 1 desplaza un byte, la fila 2 desplaza dos bytes y la fila 3 desplaza tres bytes.

Operación MixColumns

$$\begin{pmatrix} S'_{0,i} \\ S'_{1,i} \\ S'_{2,i} \\ S'_{3,i} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,i} \\ S_{1,i} \\ S_{2,i} \\ S_{3,i} \end{pmatrix}$$

Esta transformación opera sobre el estado, columna por columna para maximizar la difusión. Concretamente, cada columna i se modifica de la manera que se indica en la figura. Se trata de operaciones con polinomios, en que cada byte es considerado como un polinomio de grado 8. Cada columna se multiplica módulo x4 + 1 con el polinomio:

a(x) = {03}x3 + {01}x2 + {01}x + {02}, donde {03} = x + 1; {02} = x; {01} = 1.

Expansión de clave

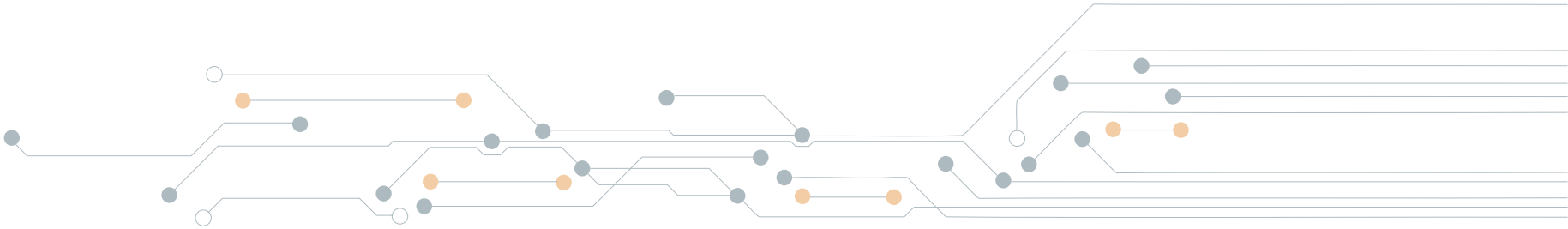
Como hemos visto, la longitud estándar de la clave en AES es de 128, 192 o 256 bits. Sin embargo, este algoritmo utiliza un total de Nr + 1 subclaves de 128 bits, es decir 16 bytes. Para ello, AES en primer lugar expande la clave mediante una serie de transformaciones hasta obtener una clave expandida de 16 (Nr + 1) bytes, a partir de la cual se obtienen las Nr + 1 subclaves.

Clave AES	Subclaves (Nr + 1)	Clave expandida
128 bits	11 de 128 bits	1.408 bits
192 bits	13 de 128 bits	1.664 bits
256 bits	15 de 128 bits	1.920 bits

Para la generación de las 10 sub claves necesarias en el caso de una cifra con AES de 128 bits, se utiliza una función denominada Expansión de Clave, que consiste en modificar la matriz de estado de la clave mediante la operación RotWord que rota el primer byte de la última palabra de 4 bytes de la matriz, aplicar luego a esa palabra resultante la operación SubBytes y sumar después or exclusivo esta palabra con la palabra que se encuentra tres posiciones atrás de esa matriz de estado y un vector conocido como Rcon, diferente para cada vuelta.

Esto dará lugar a la primera palabra de la nueva matriz de estado de la clave. Dichas operaciones se repiten 3 veces más con lo que se obtiene una matriz de estado para la clave de 4 nuevas palabras de 32 bits cada una, correspondiente en este caso a la clave de la vuelta 1. Este proceso se realiza 10 veces para obtener las 10 subclaves necesarias para cada una de las vueltas del AES 128.

El algoritmo de cifrado comienza con la función denominada AddRoundKey que realiza la suma or exclusivo entre los bytes del mensaje y los bytes de la clave. Para una clave de 128 bits, se calcularán 10 subclaves, una por cada vuelta, y se realizarán las siguientes cuatro operaciones durante nueve vueltas: SubBytes, ShiftRows, MixColumns y AddRoundKey. Para finalizar, se repiten sólo las operaciones SubBytes, ShiftRows y AddRoundKey, dando lugar a una matriz de estado final con los 16 bytes que formarán el criptograma resultado de cifrar el primer bloque de texto en claro. Para el descifrado, se recorrerá el algoritmo por el camino inverso y se usarán las funciones inversas de estas operaciones conocidas como InvSubBytes, InvShiftRows, InvMixColumns e InvAddRoundKey.

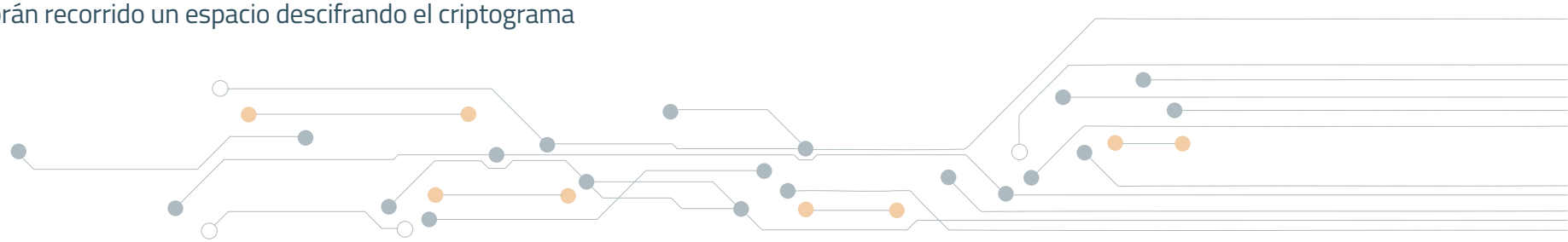


5. Ataques a los sistemas de cifra simétrica en bloque

Los sistemas de cifra simétrica en bloque deben utilizar una clave relativamente grande para evitar ataques distribuidos en red, como los sufridos por el algoritmo DES. Esto es debido a que si bien existen otros ataques a estos sistemas, más o menos sofisticados, lo cierto es que la única posibilidad real de éxito pasa por aplicar lo que se conoce como ataque por fuerza bruta, es decir descifrar el criptograma con todas las claves posibles en búsqueda de la clave verdadera. Si la clave K tiene n bits, lo normal (50% de probabilidades) es que en media debamos realizar 2n-1 intentos hasta tener éxito.

Como la clave de cifra es un único valor dentro del espacio de claves posibles, resulta muy fácil realizar un ataque distribuido en red con muchos ordenadores de forma que cada de ellos recorra un espacio de claves pequeño en búsqueda de la clave de cifra. De esta manera, si atacar una cifra simétrica con un solo ordenador y tener éxito significa un tiempo x, al usar n ordenadores simultáneamente en ese ataque, el tiempo de respuesta se reduce en n, es decir se tardará x/n. A final, n-1 ordenadores habrán recorrido un espacio descifrando el criptograma sin encontrar un texto en claro con sentido y, en cambio, un único ordenador habrá recorrido el espacio de claves donde precisamente se encontraba la clave K buscada, por lo que habrá dado con el texto en claro.

Es un ejemplo claro del principio de divide y vencerás.



Telefónica EDUCACIÓN DIGITAL