NOTAS DEL CURSO

CRIPTOGRAFÍA

Departamento de Álgebra. Universidad de Granada

Jesús García Miranda

Índice general

I Introducción. 1

I Preliminares 9

II Criptosistemas clásicos 10
1 Cifrado por sustitución . . . 13
1.1 Cifrado por sustitución monoalfabética . . . 14
2 Cifrados polialfabéticos . . . 24
2.1 Criptosistema de Vigenére . . . 27

III Introducción a la teoría de la Información 44
1 Introducción . . . 44
2 Cantidad de información . . . 44
3 Entropía . . . 47
4 Compresión de datos . . . 49
4.1 Algoritmo de Huffman . . . 51
5 Entropía condicionada . . . 53
6 Criptosistema seguro de Shannon . . . 55
7 Redundancia . . . 57

II Cifrado en flujo 59

IV Cifrados en flujo 60
0.1 Cifrados por flujo síncronos . . . 61
0.2 Cifrados por flujo asíncronos o autosincronizantes . . . 63
1 Generación de secuencias aleatorias . . . 65
2 Secuencias pseudoaleatorias . . . 66
2.1 Generadores de secuencias pseudoaleatorias . . . 68

III Cifrado por bloques simétrico 76

V Generalidades 77
1 Redes de Feistel . . . 77

2 Modos de operación para algoritmos de cifrado por bloques 79
2.1 Electronic Code Book 80
2.2 Cipher Block Chaining 81
2.3 Cipher Feed Back 82
2.4 Output Feed Back 84

VI Data Encryption Standard (DES) 86
1 Cifrado con DES 88
1.1 La función f 89
1.2 Generación de subclaves 94
1.3 Llaves débiles y criptoanálisis 96
1.4 Triple DES 98

VII Advanced Encryption Standard (AES) 99
1 Introducción 99
2 Descripción del AES 101
2.1 Proceso de cifrado 101
3 Expansión de clave 107
4 Proceso de descifrado 110
4.1 Invsubbytes 110
4.2 InvShiftrows 111
4.3 InvMixcolumn 111
4.4 InvAddroundkey 112
5 Apéndice: Cuerpos finitos. El cuerpo $\mathbb{F}_{256}$. 114
5.1 Polinomios con coeficientes en un cuerpo. 115
5.2 El cuerpo $\mathbb{F}_{256}$ 146

IV Criptografía Asimétrica 153

VIII Introducción 154
1 Requisitos de la criptografía asimétrica 155
2 Protocolo de Intercambio de Diffie-Hellman 156

IX El criptosistema RSA 158
1 Descripción del algoritmo 158
1.1 Elección de las claves 158
1.2 Cifrado y descifrado 159
1.3 Precauciones en la elección de los parámetros 162

X Criptosistema ElGammal 169
1 Descripción del algoritmo 169
1.1 Elección de las claves 169

1.2 Cifrado y descifrado 170

XI Apéndice: Fundamentos matemáticos **172**
1 Aritmética modular 172
1.1 Generalidades 172
1.2 Cálculo de inversos 174
1.3 Cálculo de potencias 177
1.4 Raíces cuadradas modulares 180
1.5 Logaritmo discreto 185
2 Tests de primalidad 190
2.1 Test de Fermat 190
2.2 Test de Solovay-Strassen 192
2.3 Test de Miller-Rabin 192
3 El problema de la factorización 196
3.1 División por tentativa 197
3.2 Método de Fermat 198
3.3 Algoritmo ρ de Pollard 198
3.4 Algoritmo de Strassen. 201
3.5 Raíz cuadrada y factorización. 203

V Aplicaciones criptográficas **205**
4 Introducción. 206

XII Autenticación y firmas digitales. **207**
1 Introducción. 207
2 MAC 209
3 Funciones Resumen 209
3.1 Generalidades 209
3.2 Función SHA1 211
4 HMAC 213
5 Firma digital RSA 214
6 Firma ElGamal 215
7 DSS 218
8 Certificados digitales. 220

XIII Protocolos criptográficos **224**
1 Introducción 224
2 Protocolo del lanzamiento de una moneda. 225
3 Protocolos de secreto compartido 226
4 Protocolos de conocimiento cero 230
4.1 Protocolo de Chaum, Evertse y Van de Graaf 230
4.2 Protocolo de Fiat-Shamir. 231

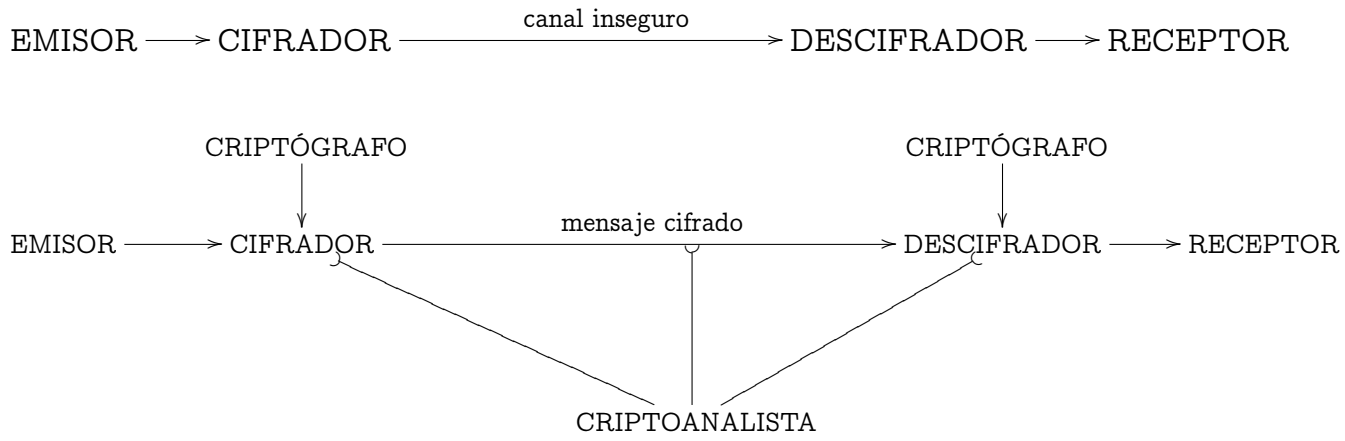
5 Protocolos de transferencia inconsciente o trascordada. 232
5.1 Protocolo de Rabin. 232
5.2 Protocolo de Berger, Peralta y Tedrick. 232
6 Protocolos de compromiso con un bit 233
6.1 Protocolo de Brassard, Crépeau y Chaum. 233

CRIPTOGRAFÍA I

Introducción.

Es tradicional comenzar un curso de *Criptografía* por un análisis sintáctico de la palabra *criptografía*. Esta palabra está compuesta por dos palabras griegas "*cryptos*", (χρυπτόσ) que significa esconder u ocultar, y "*graphein*" (γραφή) que significa escribir. Así pues etimológicamente criptografía significa escritura oculta.
Podemos decir que la *criptografía* es el arte de construir mensajes, que son ininteligibles para un observador cualquiera, y que un observador autorizado, por ejemplo, la persona o personas a las que va dirigido el mensaje, pueden interpretar.
De hecho, la RAE define *criptografía* como el arte de escribir con clave secreta o de modo enigmático.
Sin embargo, para lo que nos interesa esta definición contiene algunas imprecisiones.
Por ejemplo, la criptografía podría considerarse una ciencia, no un arte.
Se puede emplear la criptografía no sólo para escribir, sino para comunicaciones telefónicas.
Se pueden emplear y de hecho se emplean más de una clave.
Muchos sistemas criptográficos utilizan no solo una clave secreta, sino que usan también una clave pública.
Podríamos considerar entonces como definición de criptografía (ver libro electrónico de seguridad informática y criptografía, de Jorge Ramió) *Rama inicial de las Matemáticas y en la actualidad también de la Informática y la Telemática, que hace uso de métodos y técnicas con el objeto principal de cifrar, y por tanto proteger, un mensaje o archivo por medio de un algoritmo, usando una o más claves.*
La necesidad de la criptografía se entiende desde el momento en que alguien quiere transmitir información confidencial por una vía de comunicación, que se supone insegura, y hay otros que quieren recibir esta información sin que unos terceros conozcan su verdadero significado o contenido.
Hasta no hace demasiado tiempo, la interacción social se efectuaba bien cara a cara, bien mediante el teléfono analógico, bien por el correo escrito. Estas actividades pueden considerarse intrínsecamente seguras. Reconocer a los interlocutores no presenta generalmente problemas. Los documentos escritos se autentifican mediante las firmas, y las posibles falsificaciones dejan huellas físicas que pueden ser detectadas mediante diversos procedimientos.
Con la aparición del mundo digital, la interacción social se efectúa cada vez en mayor medida mediante comunicaciones digitales: correo electrónico, cajeros automáticos, tarjetas inteligentes, telefonía móvil digital, telecompra, etc. Todo ello utilizando cadenas de unos y

ceros.
Nadie duda de los avances que ha supuesto en la sociedad el desarrollo de las comunicaciones digitales. Pero junto a sus grandes ventajas, tiene también sus puntos débiles: la vulnerabilidad de los bits. Dichos bits no son seguros y no tienen personalidad (como puede tenerla una firma manuscrita), y un cambio de algunos bits en un mensaje no dejaría huella física alguna.
Por tanto, el contenido de las comunicaciones digitales puede alterarse si no se toman las medidas oportunas de protección. Estas alteraciones pueden producirse, bien por la deficiencia de los canales de comunicación, bien por la manipulación interesada de los comunicantes, o bien de terceros. La protección contra las posibles modificaciones producidas por los canales se efectúa mediante los códigos correctores de errores. La protección contra las manipulaciones intencionadas, mediante la criptografía.
Si bien la criptografía ha existido desde siempre, ha sido en los últimos años, con la aparición y desarrollo de las comunicaciones digitales cuando esta ciencia ha cobrado mayor auge, y de ser algo reservado para temas casi exclusivamente militares y diplomáticos, ha pasado a ser una ciencia de interés general.
Cualquier empresa, cualquier banco, necesita proteger determinados datos que de ser manipulados, o descubiertos por atacantes podría tener graves consecuencias para sus intereses.
Con la criptografía aparecen dos acciones que son dignas de tener en cuenta: *el cifrado* que consiste en la *construcción del mensaje oculto* y el *descifrado* que consiste en *recuperar el mensaje original a partir del mensaje oculto*.
A los que trabajan en diseñar métodos y procedimientos (algoritmos) para descubrir el mensaje oculto sin ser el cifrador o el descifrador los llamaremos *criptoanalistas*, y al proceso que desarrollan lo llamaremos *criptoanálisis*. En contraposición, llamaremos *criptógrafos* a los que diseñan algoritmos para cifrar y descifrar y *criptografía* al proceso de crear y ejecutar estos algoritmos.
Nos encontramos pues con dos "*ciencias*": la criptografía y el criptoanálisis, una encargada de construir y otra encargada de descubrir las técnicas diseñadas por la primera. En este curso nos centraremos en el estudio de la criptografía, y dejaremos el criptoanálisis para cursos más avanzados.
Representamos en el siguiente dibujo los actores de la escena que acabamos de presentar.



$$\text{EMISOR} \longrightarrow \text{CIFRADOR} \xrightarrow[\text{canal inseguro}]{\text{canal seguro}} \text{DESCIFRADOR} \longrightarrow \text{RECEPTOR}$$

Observar que en el primer gráfico hemos indicado que existen dos tipos de canales de comunicación: *seguros* e *inseguros*.
Si utilizamos un canal de comunicación seguro no tenemos necesidad de la criptografía ya que por definición no puede existir un observador, ajeno al emisor o al receptor, que pueda acceder al mensaje.
Pero, ¿existen canales de comunicación seguros?
En principio podemos suponer que sí. Un ejemplo de canal seguro podría ser la comunicación verbal directa entre dos personas o la trasmisión directa de un texto escrito.
Los canales inseguros serían aquellos a los que puede acceder un tercero. Un ejemplo sería la comunicación por teléfono, y por extensión por otros medios de comunicación electrónica.

Vamos a ver el siguiente ejemplo:
Juan y María necesitan durante un tiempo mantener una comunicación. Si pudieran verse cara a cara, no habría problema, pero durante ese tiempo van a permanecer en lugares distantes, de forma que no sea posible el contacto personal. La comunicación deberá llevarse a cabo entonces, bien a través del teléfono, bien del correo (postal o electrónico).
Sin embargo, Miguel controla todos los canales de comunicación. Tiene los teléfonos intervenidos. Los mensajes electrónicos viajan a través de una red con unos números que identifican al ordenador de origen y destino, con lo que fácilmente podría acceder a dichos mensajes.
La solución es, entonces, ocultar el mensaje de alguna forma que sea incomprensible para Miguel, pero no para Juan y María.
Las posibilidades aquí son infinitas. Por ejemplo, podrían decidir invertir el mensaje: así, para decirse HOLA, escribirían ALOH, pero esto sería fácilmente detectable. Deciden entonces, además de esa transformación, sustituir cada letra por la siguiente del alfabeto, con lo cual el mensaje quedaría BMPI. ¿Quien puede saber que eso significa HOLA?.
Con esta información no sería fácil descifrar el mensaje (salvo que se sepa la transormación que se ha usado). Pero si Juan y María intercambian muchos mensajes que son leídos por Miguel, éste no tardaría mucho en descubrir el secreto. Encontraría letras que se repiten con más frecuencia, que corresponderían con la "E" y la "A" por ser estas las de más frecuencia en Español, que además serían la F y la B. A partir de esto, con algunos mensajes sería muy sencillo conocer el método de cifrado.
Pensando un poco más se pueden ocurrir nuevas ideas. Por ejemplo, que en el proceso de transformación del mensaje intervenga algún dato como un número, una palabra, que pudiera variarse fácilmente y de tal forma que el mensaje cifrado dependa de él. Así, en el ejemplo anterior, tras invertir el orden de las letras hemos tomado la siguiente letra del alfabeto. ¿Y si dejamos libertad para elegir, en lugar de la siguiente, la que le sigue 7 posiciones, o 15, etc.?

En tal caso, el método de cifrado depende de un número que en el ejemplo antes analizado vale 1, pero si valiera 3, el mensaje HOLA, cifrado, sería ahora DÑRK. Dicho dato es lo que se conoce como *clave*

Para proceder de forma eficaz, lo que hacemos en primer lugar es considerar cada mensaje como una secuencia de signos. Estos pueden ser letras (mayúsculas o minúsculas), números, signos de puntuación, etc. Normalmente no es necesario emplearlos todos. Por ejemplo, podemos no distinguir entre mayúsculas y minúsculas, podemos olvidarnos de los espacios, de los signos de puntuación.

Lo que hacemos, en definitiva, es fijar un alfabeto con el que vamos a construir los mensajes. Este alfabeto se denomina *alfabeto en claro o en llano*, y a los textos con él construidos *texto en claro o llano (del inglés plain text)*

Estos textos en claro, lo que vamos a hacer es transformarlos. Una vez transformado, tendremos lo que se llama *texto cifrado o criptograma.* El texto cifrado puede estar escrito con el mismo conjunto de signos que el texto en claro, pero no tiene porqué ser así. Por ejemplo, podemos escribir criptogramas usando sólo dígitos, o empleando nuevos símbolos a los que daremos algún significado.

A continuación hay que elegir las posibles claves. El conjunto de claves debe ser lo suficientemente grande para que, aún conociendo el método de cifrado, no se pueda, sin el conocimiento de la clave, obtener el texto llano a partir del criptograma.

Vamos a ver un ejemplo sencillo en el que el texto cifrado va a consistir únicamente de dígitos de 0 a 9.

Una clave va a ser una palabra. Elegimos, por ejemplo, la palabra APROBAR, y escribimos las letras del alfabeto, y el punto como siguen:

	1	2	3	4	5	6	7	8	9	0
			A	P	R	O	B	.	C	D
1	E	F	G	H	I	J	K	L	M	N
2	Ñ	Q	S	T	U	V	W	X	Y	Z

Para cifrar un texto, cada una de las letras del alfabeto la sustituimos por una o dos cifras, según estén en la primera, o en la segunda y tercera fila de la tabla. Así, la R se cifrafía como 5, mientras que la G como 13. Puesto que los números 1 y 2 no se corresponden con ningún signo, no hay lugar a confusión. Por ejemplo, el texto llano "ESTOY ESTUDIANDO CRIPTOGRAFIA" quedaría cifrado como sigue:

112324629811232425015310068951524613531215 3

Y a la hora de descifrarlo, tenemos que separarlo por signos, teniendo en cuenta que cuando encontremos un 1 o un 2, hemos de considerar la cifra siguiente.

11 23 24 6 29 8 11 23 24 25 0 15 3 10 0 6 8 9 5 15 24 6 13 5 3 12 15 3

Y ahora, sabiendo que 11 se corresponde con A, 23 con S, 24 con T, 6 con O, etc. se puede descifrar.

Sin embargo, con este método de cifrado nos encontramos con el mismo inconveniente que teníamos antes. La E siempre se cifra como 23, la A como 3, así que un análisis de frecuencias podría desvelar la clave a algún intruso.
Así que tratamos de añadirle algo más de complejidad. Para eso, con la clave que teníamos (APROBAR) le asociamos a cada letra un número siguiendo el orden alfabético. Así, tendríamos

A	P	R	O	B	A	R
1	5	6	4	3	2	7

Y repetimos esta secuencia de dígitos hasta obtener una cadena del tamaño del texto cifrado. Una vez hecho esto, la colocamos debajo y las sumamos, sin tener en cuenta el acarreo (es decir, sumamos módulo 10), y el resultado final podría ser el texto cifrado. Veámoslo.

$$\begin{array}{r} 11232462981123242501531006895124613531215 3 \\ 15643271564327156432715643271564327156432 7 \\ \hline 26875633445440398933246649066688930687647 0 \end{array}$$

Y ahora, para hacerlo más legible, podemos agrupar en bloques de 5

26875 63344 54403 98933 24664 90666 88930 68764 70

Obsérvese como ahora, la E inicial se ha cifrado como 26, mientras que la E que es inicio de la segunda palabra se ha cifrado como 54.
Podemos ver ahora como afecta la clave al resultado final. Por una parte, provoca una distribución diferente de las letras en el rectángulo, y por otra parte, la cantidad a sumar varía. Así, el cambio de una clave da lugar a que un mismo texto se cifre de forma diferente. El mantener una clave durante mucho tiempo puede comprometer la información que se esconde en los mensajes, así que lo ideal es cambiar periódicamente las claves. Esto depende de la cantidad de mensajes que vayan a intercambiar, de la importancia que le den a que descifren su información, etc. De esta forma, pueden acordar, por ejemplo, cambiar la clave cada 3 días, para lo cual, confeccionan una lista de claves con claves suficientes para el tiempo que van a mantener este tipo de comunicación. Esta lista de claves la mantienen en secreto, y destruyen aquellas que ya hayan utilizado.
Este ejemplo sencillo puede servir para describir lo que sería un criptosistema.
Por una parte tenemos un conjunto de posibles mensajes o textos planos (que serían todos los posibles mensajes que pueden transmitirse Juan y María), que llamaremos $\mathcal{M}$
Además, tenemos todos los posibles textos cifrados o criptogramas (en este caso, determinadas cadenas de cifras), que llamaremos $\mathcal{C}$
Tenemos un conjunto de claves K, que son las palabras de menos de 10 letras (podríamos elegir palabras con significado, o simplemente una sucesión de 9 o menos letras)
Para cada $k \in K$, una función $E_k : \mathcal{M} \to \mathcal{C}$, que llamaremos función de cifrado.
Para cada $k \in K$, una función $D_k : \mathcal{C} \to \mathcal{M}$, que llamaremos función de descifrado. Estas dos funciones están relacionadas mediante la ecuación $D_k \circ E_k = Id_{\mathcal{M}}$ (es decir, el descifrado de un texto cifrado es el texto de partida).

Este ejemplo sirve para dar una definición de lo que es un criptosistema abstracto.
Definición:
Definimos un Criptosistema abstracto como un quinteto $\mathbb{S} = (\mathcal{M}, \mathcal{C}, K, E, D)$ donde:

- $\mathcal{M}$ es el conjunto de mensajes sin cifrar (texto plano)
- $\mathcal{C}$ es el conjunto de mensajes cifrados (criptogramas)
- K es un conjunto de claves.
- E es una aplicación $E : \mathcal{M} \times K \to \mathcal{C}$. Para un elemento $k \in K$ fijo, la aplicación $\mathcal{M} \to \mathcal{C}$ dada por $m \mapsto E(m, k)$ la denotaremos como $E_k : \mathcal{M} \to \mathcal{C}$.
- D es una aplicación $D : \mathcal{C} \times K \to \mathcal{M}$. Para un elemento $k \in K$ fijo, la aplicación $\mathcal{C} \to \mathcal{M}$ dada por $c \mapsto D(c, k)$ la denotaremos como $D_k : \mathcal{C} \to \mathcal{M}$.

Además, para cada $k \in K$, y cada $m \in \mathcal{M}$ se verifica que $D_k(E_k(m)) = m$
Los conjuntos $\mathcal{M}$, $\mathcal{C}$ y K son finitos, y en ocasiones pueden coincidir $\mathcal{M}$ y $\mathcal{C}$.

Ya hemos comentado antes de la necesidad, en nuestros días, de proteger determinados datos. Actualmente, el uso de internet ha revolucionado el mercado y la sociedad. El comercio electrónico está avanzando a una velocidad vertiginosa. Las empresas utilizan la red para todo tipo de intercambio y almacenamiento de información. A partir de este uso masivo de las comunicaciones surgen nuevos problemas, y por tanto es necesario desarrollar nuevos sistemas de protección para los que la criptografía es la herramienta más adecuada. Cualquiera de nosotros dispone de una tarjeta de crédito para realizar sus transacciones bancarias.
En la propia Universidad una gran cantidad de datos viajan a través de Internet: datos personales, expedientes, etc. La manipulación, o incluso sólo el acceso por parte de alguien malintencionado a dichos datos podría ocasionar importantes daños.
Vamos a enumerar a continuación algunos aspectos en los que la criptografía juega un papel importante, y algunos problemas que puede resolver.

- Permite garantizar la autenticidad de origen de un documento digital, es decir, que el documento proviene de la persona o entidad que dice ser. En los documentos escritos esto se garantiza mediante la firma manuscrita.
- Permite garantizar la integridad del documento, es decir, que no ha sido modificado. Ya hemos comentado previamente que un documento es una sucesión de bits, y en principio, un cambio en algunos de éstos no dejaría ninguna huella física, al contrario de una manipulación en documentos manuscritos.
- Permite evitar la repudiación de los mensajes por parte de los comunicantes. Esto podría ocurrir cuando, bien el emisor, bien el receptor de un determinado mensaje intente negar la emisión o recepción del mismo. En los documentos manuscritos esto se evitaba bien realizando el envío a través de un correo certificado, con acuse de recibo, bien mediante la firma en presencia de testigos.

- Permite evitar el problema de la falta de simulteneidad en las firmas digitales de contratos. Podría darse el caso de falta de confianza y que una de las partes se niegue a firmar un contrato hasta asegurarse que la otra parte lo ha hecho.
- Permite verificar la identidad de los comunicantes. Es decir, si cada comunicante es quien realmente dice ser.

Lo dicho hasta ahora creemos que justifica que la criptografía sea una disciplina con la suficiente entidad para ser incluida en el curriculum de los estudiantes de Ingeniería Informática, así como de los de Ingeniería de Telecomunicaciones. En estas notas trataremos de dar una introducción a algunos conceptos criptográficos

Parte I

Preliminares

CRIPTOGRAFÍA II

Criptosistemas clásicos

En este capítulo vamos a dar un breve repaso a la historia de la criptografía.
Prácticamente se puede afirmar que la criptografía es casi tan antigua como la propia escritura. Sin embargo, la documentación que existe sobre su empleo en las civilizaciones antiguas se reduce a un uso ocasional. El uso regular de la criptografía comienza sobre la Edad Media. No obstante, vamos comenzar nuestro estudio por las civilizaciones antiguas.
Así, el primer texto de Criptografía del que se tiene conocimiento podemos situarlo sobre el 1900 a.C., en el Antiguo Egipto. Sin embargo parece ser que el propósito no es ocultar el contenido de un mensaje, sino que, quizá para dotar al texto de un cierto tono de dignidad, se cambian ciertos símbolos jeroglíficos por otros similares pero poco usados.
En la antigua Mesopotamia se empleaba una técnica similar: cambiar los signos cuneiformes de su escritura por otros, pero ahora sí con la intención de ocultar el significado. El texto cifrafo más antiguo que se conserva corresponde a esta cultura, y puede tener una antigüedad de unos 3500 años. En él se detallaba una fórmula para un barniz que se usaba en alfarería.
De esta época son también conocidos otras técnicas para ocultar mensajes. Por ejemplo, se tomaba un esclavo, se le rapaba la cabeza y se le tatuaba en ella un mensaje. Una vez le había crecido el cabello, el esclavo llevaba el mensaje a su destinatario, que no tenía más que raparlo para leer el mensaje.
En el siglo VI a.C nos encontramos con las conocidas como cifras hebraicas, que eran empleados por los hebreos fundamentalmente en textos bíblicos. Las más conocidas son "Atbash", "Albam" y "Atbah", que aparecen en el Libro de Jeremías. Consisten en sustituciones de unos caracteres por otros. Así, el Atbash, la primera letra del alfabeto hebreo (Aleph) se sustituye por la última (Taw), la segunda, Beth, por la penúltima (Shin), y así sucesivamente. En el Albam, la sustitución es, la primera por la décimosegunda (Lamed), la segunda por la décimotercera (Mem). El tercero, el Atbah, aunque es también una sustitución, no sigue una regla tan simple como las anteriores. Aleph es reemplazada por Teth (la novena), Beth es reemplazada por Mem (décimo segunda). No se conoce, no obstante, razón alguna para tal cifrado.
La diplomacia China usaba un método que consistía en escribir el mensaje en una fina seda que se enrollaba y se sellaba con cera. El mensajero, lo ocultaba en su propio cuerpo, tragándoselo.
Según Herodoto, un exiliado griego en Persia, allá por el siglo V a.C. grabó los planes persas en un par de tablillas de madera, y después las cubrió con cera. De esta forma el mensaje quedaba oculto. Estas tablillas fueron transportadas desde la ciudad persa de Susa hasta

Esparta. Al llegar a Esparta, la esposa del Rey Leónidas se percató de que bajo la cera había algo escrito. No tuvieron más que derretir la cera, y el mensaje quedó para ser leído. Gracias a esto, los griegos tuvieron conocimiento de los planes persas, lo que les fue muy valioso para derrotarlos en las batallas de las Termópilas, Salamina y Platea.

Los espartanos, en ese mismo siglo V a.C. diseñaron un criptosistema para uso militar, conocido como el escítalo.

Consistía en un cilindro en el que se enrollaba una cinta de pergamino larga y estrecha. Sobre ella, se escribía en forma longitudinal un mensaje, y se desenrollaba. Entonces aparecían las letras del mensaje pero en otro orden, sin ningún sentido.

Para recuperar el mensaje bastaba enrollar el pergamino en un cilindro de igual diámetro que el usado para escribirlo.

Plutarco lo describe de este modo:

> Cuando un general parte para una expedición de tierra o mar, los éforos (ministros) toman dos bastones redondos, perfectamente iguales en longitud y grosor, de manera que se correspondan exactamente uno con otro en todas sus dimensiones. Ellos guardan uno de estos bastones, dando el otro al general, y llaman a estos bastones escítalos. Cuando quieren enviar al general un secreto de importancia, cortan una tira de pergamino, larga y estrecha como una correa, arrollándola alrededor del escítalo que guardaron, sin dejar el menor intervalo entre los bordes de la banda, de tal suerte que el pergamino cubra enteramente la superficie del bastón. Sobre este pergamino así arrollado alrededor del escítalo, escriben lo que desean y después quitan la cinta y la envían al general sin el bastón. El general

que la recibe no sabría leerla, porque las letras, perdida la alineación y dispersas, no tendrían continuidad; pero él toma el escítalo que llevó consigo, y arrollando alrededor la banda del pergamino, se reunirán las vueltas, volviendo las letras a tormar el primitivo orden en que fueron escritas. Esta misiva se llama escítalo, del nombre mismo del bastón, como aquello que se mide toma el nombre de aquello que le sirve de medida.

También dentro de la civilización griega, aunque 3 siglos más tarde, Polibios describe un método que consiste en disponer las letras en un cuadrado

	1	2	3	4	5
1	α	β	γ	δ	ϵ
2	ζ	η	θ	ι	κ
3	λ	μ	ν	ξ	o
4	π	ρ	σ	σ	τ
5	υ	φ	χ	ψ	ω

y así cada letra era representada por un par de números. Por ejemplo, π era representado como 41.

La información se transmitía mediante señales luminosas con antorchas.

Aparece aquí una idea muy importante y que va a estar presente a lo largo de toda la historia de la criptografía, como es la de representar las letras por números. Luego, esto permitirá aprovechar la aritmética de los números para obtener sistemas criptográficos más complejos.

Muy conocido es el método ideado por Julio Cesar. Consiste en sustituir cada letra del abecedario latino por la que se encuentra tres posiciones más avanzadas según el orden alfabético.

Curiosamente, una de las descripciones más antiguas de cifrado la encontramos en el Kamasutra, un texto escrito el siglo IV d.C. por el sabio hindú Vatsyayana, sin embargo basado en manuscritos datados de más de 800 años. El Kama-Sutra recomienda que las mujeres estudien 64 artes, incluyendo la culinaria, la forma de vestir, masaje y la preparación de perfumes. La lista también incluye algunos artes menos obvios como prestidigitación, ajedrez, encuadernación de libros y carpintería. En la lista, la número 45 es la *mlecchita-vikalpa*, el arte de la escritura secreta, indicada para ayudar a las mujeres a esconder los detalles de sus relaciones amorosas. Una de las técnicas recomendadas es la de formar pares aleatorios de letras del alfabeto y después sustituir cada letra del texto original por la correspondiente en el par.

Si analizamos las diferentes técnicas aquí descritas, podemos clasificarlas en tres grupos.

En primer lugar tenemos uno conocido como *esteganografía*. En este, el mensaje no se modifica, sino que simplemente se oculta su existencia (tablas de cera, esclavo). Otro muy empleado es el de escribir con tinta invisible. Al entrar en contacto con ciertas sustancias, o con calor, la tinta se hace visible y se puede leer el mensaje. La esteganografía como tal, no se considera como parte de la criptografía, aunque es indudable que guarda una relación con ella.

Los otros dos grupos, que sí podemos considerar dentro de la criptografía serían los denominados *Cifrado por transposición* y *Cifrado por sustitución.*

En el primero (cifrado por transposición), lo que se hace es cambiar de orden las letras o caracteres del texto llano, siguiendo un criterio acordado por el emisor y el receptor (un ejemplo de esto es el escítalo).

En el segundo (cifrado por sustitución), cada carácter del texto llano se sustituye por otro (que puede ser del mismo alfabeto -cifras hebraicas, cifrado de Cesar, Kamasutra- o de otro diferente -Polibios-)

La diferencia fundamental entre criptografía y esteganografía (del griego steganos -escondido- y grafia -escritura-) es que en la primera no se pretende ocultar la existencia de un mensaje, sino hacerlo ininteligible para determinadas personas, lo cual supone cambiar la forma de escritura, mientras que la esteganografía pretende ocultar la existencia del mensaje (no es necesario cambiar el mensaje, sino esconderlo).

1 Cifrado por sustitución

Como hemos dicho es el cifrado que consiste en cambiar cada signo del texto llano por otro símbolo, del mismo o de otro alfabeto.

Los cifrados por sustitución pueden clasificarse en cifrados por sustitución simple (o monoalfabética) y cifrados por sustitución múltiple (o polialfabética).

Los ejemplos que nos han salido de cifrados por sustitución son todos monoalfabéticos, y el más sencillo es el criptosistema de Cesar.

Este se corresponde como lo que se llama *cifrado por traslación*, que es un caso particular del cifrado por permutación

. . . . . 1.1

Cifrado por sustitución monoalfabética

Cifrado por permutación

En el cifrado por permutación, se emplea el mismo alfabeto para texto llano que para texto cifrado. Dicho alfabeto debe estar ordenado, y por tanto, si el alfabeto tiene m elementos, cada elemento de dicho alfabeto puede ser considerado como un número entre 0 y $m-1$.
En el caso de que consideremos como alfabeto el nuestro, la correspondencia sería

A	B	C	D	E	F	G	H	I	J	K	L	M	N
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
14	15	16	17	18	19	20	21	22	23	24	25	26

Es decir, que, bajo esa correspondencia, nuestro alfabeto podría ser $\mathbb{Z}_{27}$.
Para el cifrado por traslación, lo único que hay que hacer es sumar una cantidad fija k a cada carácter, es decir, tenemos como función de cifrado $E_k(m) = m + k$. La función de descifrado es $D_k(c) = c - k$.
El caso del criptosistema de Cesar se corresponde con un cifrado por traslación en el que $k = 3$.
Por ejemplo, supongamos que queremos cifrar el mensaje "LA VIDA ES BELLA", y tomamos $k = 7$. Eliminando los espacios, y con la correspondencia anterior, el mensaje a cifrar es

11 0 22 8 3 0 4 19 1 4 11 11 0

y para cifrarlo, sumamos 7 a cada una de las cantidades que tenemos (y reducimos módulo 27)

18 7 2 15 10 7 11 26 8 11 18 18 7

luego el texto cifrado es

RHBOKHLZILRRH

Para descifrar el mensaje, si no conocemos k, podemos realizar lo que se conoce como un ataque por fuerza bruta, que consiste en probar con todos los posibles valores de k, hasta encontrar un mensaje con sentido. Así, si $k = 0$, el mensaje llano sería RHCOKHLZILRRH, que no tiene sentido.
Si $k = 1$, el texto llano sería QGBÑJGKYHKQQG
Si $k = 2$, entonces tendríamos PFANIFJXGJPPF
Si $k = 3$, entonces el texto llano es OEZMHEIWFIOOE

Si $k = 4$, tenemos ÑDYLGDHVEHÑÑD
Si $k = 5$, entonces NCXKFCGUDGNNC
Si $k = 6$, el texto llano sería MBWJEBFTCFMMB
Si $k = 7$ tendríamos LAVIDAESBELLA
luego ya hemos descifrado el texto.
No obstante, para textos más largos podemos emplear una técnica que se conoce como análisis de frecuencias. Esta se basa en que la distribución de letras en un idioma no es uniforme. Así, en español, la letra más frecuente es la E, con una frecuencia aproximada del 14 % (obviamente, resulta imposible estudiar todos los textos españoles, así que según los que se hayan utilizado para el estudio, la frecuencia puede variar ligeramente)
Para el español vamos a utilizar el siguiente cuadro de probabilidades de aparición de una letra.

letra	prob.	letra	prob.	letra	prob.
E	0,13626	C	0,04662	Q	0,00872
A	0,12484	T	0,04612	H	0,00701
O	0,08653	U	0,03920	F	0,00691
S	0,07951	M	0,03139	Z	0,00521
R	0,06848	P	0,02496	J	0,00441
N	0,06688	B	0,01415	Ñ	0,00234
I	0,06226	G	0,01002	X	0,00220
D	0,05835	Y	0,00892	W	0,00023
L	0,04953	V	0,00892	K	0,00003

Por ejemplo, consideremos el siguiente criptograma, que sabemos que se corresponde con un texto español que ha sido cifrado mediante un cifrado por traslación
ÑWEWUEPKBNÑUKVKWMQKNÑMEIYWYVLBÑWYAERÑBY
KMYBNKBVÑWYQKVEMQYDRÑVZYAEÑFRFRKEWQRNKU
PYNÑUYCNÑUKWJKÑWKCDRUUÑBYKNKBPKKWDRPEKB
YMRWOUKMYIPKUPYMYBBÑNYB
Podríamos intentar probar con las 27 claves, y descifrando con cada una de ellas encontrar cual tiene sentido, pero lo que vamos a hacer ahora es contar cuantas veces aparece cada letra, y obenemos así la siguiente tabla:

K	19	R	8	C	2
Y	16	M	7	F	2
Ñ	13	P	6	I	2
B	11	V	5	J	1
W	11	Q	4	L	1
U	9	D	3	O	1
E	8	A	2	Z	1
N	8				

Puesto que en español la letra más frecuente es la E, suponemos que la E se ha cifrado como K. Esto nos da $k = 6$ (es decir, traslación de 5 posiciones). En tal caso, la A se habrá cifrado

como G. Pero vemos que la G no aparece, lo que nos diría, caso de ser $k = 6$, que la A no aparece en el texto llano, lo cual parece poco probable.

Probamos entonces con la Y como la letra que cifra la E, en cuyo caso $k = 21$. Eso nos daría que la A se cifra como U (que aparece 9 veces), pero la O, la tercera más frecuente se cifra como J, que aparece sólo una vez.

Caso de cifrarse la E como Ñ, nos daría una clave $k = 10$, en cuyo caso, la A se cifraría como K (que aparece 19 veces) y la O como Y (que aparece 16 veces), es decir, aunque en orden diferente aparecen las tres letras más frecuentes del español. Por tanto, suponemos que la clave es 10, y desciframos, obteniendo:

ENUNLUGARDELAMANCHADECUYONOMBRENOQUIERO
ACORDARMENOHAMUCHOTIEMPOQUEVIVÍAUNHIDALGO
DELOSDELANZAENASTILLEROADARGAANTIGUAROCIN
FLACOYGALGOCORREDOR

que vemos se corresponde con un fragmento de una importante obra de la literatura española.

En realidad, la función de cifrado de este criptosistema una biyección $E_k : \mathbb{Z}_{27} \to \mathbb{Z}_{27}$ cuya inversa es D_k.

Dado un texto $m = x_1x_2\cdots x_n$, donde cada x_i es un elemento de $\mathbb{Z}_{27}$ (o una letra del alfabeto español), su cifrado, que denotaremos por $E_k(m)$, es

$$E_k(m) = E_k(x_1)E_k(x_2)\cdots E_k(x_n)$$

Pero la biyección E_k es una biyección muy particular, pues conocida la imagen de un elemento de $\mathbb{Z}_{27}$, tenemos determinada toda la aplicación.

¿Qué nos impide eliminar esa restricción?.

Es decir, en lugar de tomar únicamente las biyecciones $\mathbb{Z}_{27} \to \mathbb{Z}_{27}$ de la forma $x \mapsto x + k$, por qué no tomamos como posibles claves todas las biyecciones de $\mathbb{Z}_{27}$ en sí mismo. Ese conjunto, sabemos que es el grupo simétrico S_{27}, que tiene 27! elementos.

Entonces, para cada elemento $\sigma \in S_{27}$ y para cada texto a cifrar $m = x_1x_2\cdots x_n$ consideramos la función E_σ definida por:

$$E_\sigma(m) = \sigma(x_1)\sigma(x_2)\cdots\sigma(x_n).$$

La función de descifrado D_σ está definida por σ^{-1}.

Recordemos que la existencia de σ^{-1}, la inversa de σ, es consecuencia de que S_{27} es un grupo.

Obtenemos así lo que sería un cifrado por permutación. Lo que hace es permutar (reordenar) los caracteres del texto llano.

Por ejemplo, consideramos el siguiente elemento de S_{27}:

$$(0\ 16\ 18\ 10\ 25\ 6\ 15)(1\ 2\ 21\ 20\ 8\ 13\ 23\ 4\ 12)(3\ 26\ 14\ 9\ 19\ 22\ 7\ 17\ 24\ 11)$$

que con la correspondencia entre letras y números viene a ser la sustitución

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

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
J	A	R	X	K	V	I	T	H	E	L	G	Ñ

En tal caso, el cifrado del texto llano

OBTENEMOSASILOQUESERIAUNCIFRADOPORPERMUTACION
LOQUEHACEESPERMUTARLOSCARACTERESDELTEXTOLLANO

nos quedaría

XCIHWHBAVPVNDAXTHVHKNPTWUNFKPZARAKRHKBTIPUNAW
DAXTMQPUMMVRHKBTIPKDAVUPKPUIHKHVZHDIMLIADDPWA

Vamos a criptoanalizar un texto que sabemos que ha sido cifrado mediante una permutación. Sea el criptograma siguiente.

GEVWROFRBETEWRTEWAIESCRGPORRÑNVEÑAÑWR
VROGRVPÑPORFRLEÑHPVTCBEVRÑWRVTAÑPOFRR
OWVCIWCVEOUOCOAÑWRVEIIAPÑROIPTPGCRFRO
RVRBIEOPFRBEWRPVAEFRNVCGPOHAÑAWPOPBEW
RPVAEEÑEBAWAIEFRÑCTRVPO

En este caso, si intentamos romper el cifrado probando con las distintas claves, es decir, con los distintos elementos de S_{27} tendríamos que probar con un total de

$$27! = 10{,}888_4869{,}450_3418{,}352_2160{,}768_1000{,}000 \sim 10^{28}$$

posibles claves, luego este método no es viable.
Vamos a emplear entonces el criptoanálisis basado en el análisis de frecuencias para intentar romperlo.
Al ser demasiado pequeño, un análisis de frecuencias puede ser costoso, así que daremos la información adicional de que el texto está relacionado con las matemáticas.
En nuestro texto las frecuencias de las distintas letras es:

letra	frec.	letra	frec.	letra	frec.	letra	frec.
A	11	I	7	P	15	X	0
B	6	J	0	Q	0	Y	0
C	8	K	0	R	27	Y	0
D	0	L	1	S	1		
E	18	M	0	T	6		
F	7	N	2	U	1		
G	5	Ñ	12	V	15		
H	2	0	15	W	12		

Comencemos a descifrar el texto cifrado; el único método es el de ensayo y error, eso sí, teniendo en cuenta las probabilidades de las letras en el idioma español y las frecuencias en nuestro texto cifrado.

Suponemos que la letra R cifra a la letra E y que la letra E cifra a la letra A. Y sustituimos.

GaVWeOFeBaTaWeTaWAIaSCeGPOeeÑNVaÑAÑWe
VeOGeVPÑPOeFeLaÑHPVTCBaVeÑWeVTAÑPOFee
OWVCIWCVaOUOCOAÑWeVaIIAPÑeOIPTPGCeFeO
eVeBIaOPFeBaWePVAaFeNVCGPOHAÑAWPOPBaW
ePVAaaÑaBAWAIaFeÑCTeVPO

Las siguientes letras en frecuencia son O, P y V, que puede cifrar a O, S, R, N, I que son las siguientes letras en probabilidad en el idioma español. Hacemos, por ejemplo la asignación: $O \mapsto S$, $P \mapsto O$ y $V \mapsto N$ (opción 1) ó R (opción 2).

Opción 1

GanWesFeBaTaWeTaWAIaSCeGoseeÑNnaÑAÑWe
nesGenoÑoseFeLaÑHonTCBaneÑWenTAÑosFee
sWnCIWCnasUsCsAÑWenaIIAoÑesIoToGCeFes
eneBIasoFeBaWeonAaFeNnCGosHAÑAWosoBaW
eonAaaÑaBAWAIaFeÑCTenos

o bien

Opción 2

GarWesFeBaTaWeTaWAIaSCeGoseeÑNraÑAÑWe
resGeroÑoseFeLaÑHorTCBareÑWerTAÑosFee
sWrCIWCrasUsCsAÑWeraIIAoÑesIoToGCeFes
ereBIasoFeBaWeorAaFeNrCGosHAÑAWosoBaW
eorAaaÑaBAWAIaFeÑCTeros

Las siguientes letras en frecuencia son A, Ñ y W, que puede cifrar a I, L, (N ó R) o cualquiera de la columna U, D, C, T,... y así seguimos el proceso.

Pero, por ejemplo, si suponemos que la palabra *MATEMATICA* aparece en el texto original, un buen lugar para aparecer en el texto cifrado sería en el segmento *TaWeTaWAIa*, que se encuentra en la primera línea. Entonces la T cifra a la letra M, la W a la letra T, A a la letra I e I a la letra C; en este caso obtenemos, siguiendo la opción 2:

Opción 2

GartesFeBamatematicaSCeGoseeÑNraÑiÑte
resGeroÑoseFeLaÑHormCBareÑtermiÑosFee
strCctCrasUsCsiÑteraccioÑescomoGCeFes
ereBcasoFeBateoriaFeNrCGosHiÑitosoBat
eoriaaÑaBiticaFeÑCmeros

Nos falta por localizar la letra N; la letra de mayor frecuencia en este momento es la Ñ; supongamos que cifre a la letra N, entonces tenemos:

Opción 2
GartesFeBamatematicaSCeGoseenNraninte
resGeronoseFeLanHormCBarenterminosFee
strCctCrasUsCsinteraccionescomoGCeFes
ereBcasoFeBateoriaFeNrCGosHinitosoBat
eoriaanaBiticaFenCmeros

Nos falta una vocal por localizar; ésta debe ser una de las de mayor frecuencia de las que nos quedan. Probemos con la C

Opción 2
GartesFeBamatematicaSueGoseenNraninte
resGeronoseFeLanHormuBarenterminosFee
structurasUsusinteraccionescomoGueFes
ereBcasoFeBateoriaFeNruGosHinitosoBat
eoriaanaBiticaFenumeros

Ahora hacemos los cambios $G \mapsto P$, $F \mapsto D$, $B \mapsto L$, que son las de mayor frecuencia.

Opción 2
partesdelamatematicaSueposeenNraninte
resperonosedeLanHormularenterminosdee
structurasUsusinteraccionescomopuedes
erelcasodelateoriadeNruposHinitosolat
eoriaanaliticadenumeros

Y ya podemos probar con los cambios $S \mapsto Q$, $N \mapsto G$, $H \mapsto F$, $U \mapsto Y$, $L \mapsto J$.

Opción 2
partesdelamatematicaqueposeengraninte
resperonosedejanformularenterminosdee
structurasysusinteraccionescomopuedes
erelcasodelateoriadegruposfinitosolat
eoriaanaliticadenumeros

Finalmente hemos logrado descifrar el texto. Observemos que para poder hacerlo hemos hecho acopio de información extra al texto cifrado, en este caso hemos utilizado que los textos estándar en español tienen una distribución de letras más o menos predeterminada, y que el texto tenía relación con las matemáticas. Es evidente que si el texto a cifrar no fuese un texto estándar, léase un texto científico sobre micología, por ejemplo, entonces la distribución de las letras sería diferente, y el estudio del descifrado sería más arduo y complicado.

Observar que este ejemplo nos enseña algo que va a ser fundamental en lo que sigue. ¿Qué ocurriría si el texto a cifrar no fuese un texto estándar? Por ejemplo, si el texto hubiese sido obtenido por algún proceso que aún desconocemos, y que podemos revertir, a partir de un texto estándar

Es claro que en este caso no sería posible aplicar el criptoanálisis que acabamos de usar, ya que no sería de aplicación toda la información adicional que hemos usado. Así pues de cara a crear un criptosistema seguro deberemos tratar de utilizar esto que aquí hemos aprendido.

Acabamos de ver como criptoanalizar un criptograma obtenido mediante un cifrado por sustitución. Ejemplos de tales cifrados, de los que ya nos han salido son, además del Cifrado de Cesar, las cifras hebraicas, o la descrita en el Kamasutra.
Hemos visto como este tipo de cifrados no resultan seguros hoy en día. Pero durante muchos siglos, la sustitución monoalfabética había permanecido invulnerable debido al elevado número de posibles claves.
Fueron los árabes los que inventaron el criptoanálisis, es decir, la ciencia de descifrar mensajes sin conocer la clave.
Para que se pudiera inventar el criptoanálisis debían darse una serie de condiciones que se dieron en la civilización árabe. Era necesario alcanzar cierto nivel en diversas disciplinas como las mátemáticas, la estadística y la lingüística. No se sabe con exactitud quien fue el primero en darse cuenta de que la variación en la frecuencia de las letras podía explotarse para descifrar textos, pero la descripción más antigua que se conoce data del Siglo IX, a cargo del científico Al Kindi. Él fue autor de 290 libros de astronomía, medicina, matemáticas, lingüística y música. Su tratado *Sobre el desciframiento de mensajes criptográficos* incluye detallados debates sobre estadística, fonética y sintaxis árabe. En uno de sus párrafos dice:

> Una manera de resolver un mensaje cifrado, si sabemos en qué lengua está escrito, es encontrar un texto llano diferente escrito en la misma lengua y que sea lo suficientemente largo para llenar alrededor de una hoja, y luego contar cuántas veces aparece cada letra. A la letra que aparece con más frecuencia la llamamos *primera*, a la siguiente en frecuencia la llamamos *segunda*, a la siguiente *tercera*, y así sucesivamente, hasta que hayamos cubierto todas las letras que aparecen en la muestra de texto llano.
>
> Luego observamos el texto cifrado que queremos resolver y clasificamos sus símbolos de la misma manera. Encontraremos el símbolo que aparece con más frecuencia y lo sustituimos con la forma de la letra *primera* de la muestra de texto llano, el siguiente símbolo más corriente lo sustituimos por la forma de la letra *segunda*, y el siguiente por la forma de la letra *tercera* y así sucesivamente, hasta que hayamos cubierto todos los símbolos del criptograma que queremos resolver.

Una vez que se fue viendo que la sustitución monoalfabética no era un método de cifrado seguro, hubo que buscar nuevas fórmulas para proteger los mensajes.
Se abren así dos caminos, que dan lugar, por una parte, a las sustituciones homofónicas y por otra parte, a las sustituciones polialfabéticas.

Cifrado por sustitución homofónica

Si el problema de la sustitución por permutación es que no resiste al criptoanálisis basado en un análisis de frecuencias, se propone, para las letras más frecuentes usar varios signos

posibles por los que sustituir. Obviamente, es necesario añadir nuevos signos al alfabeto de cifrado, dando lugar a la *sustitución homofónica*

La primera vez que se tiene constancia del uso de este método de cifrado es en 1401, en una correspondencia que mantuvieron el duque de Mantua y Simeone de Crema. Para el alfabeto cifrado se añaden 12 signos más que se corresponden con las vocales "a", "e", "o" y "u". De esta forma, cada una de estas 4 vocales dispone de 4 caracteres del alfabeto cifrado para sustituirla.

A los distintos signos que se eligen para reemplazar la misma letra de texto plano se les denomina signos homófonos.

Si el número de signos homófonos que elegimos para cada letra lo elegimos de forma que sea proporcional a la frecuencia de aparición de dicha letra en la lengua en que se escribe, al final tendremos un criptograma con una distribución más o menos uniforme de signos, lo cual hace inviable un análisis de frecuencias. Sin embargo, si disponemos de una cantidad grande de texto cifrado, es posible realizar un criptoanálisis.

Si hay palabras que se repiten en el texto cifrado, parte de las palabras (las correspondientes a caracteres que no tienen signos homófonos) se cifran de la misma forma, mientras que los símbolos intercalados corresponderan a un mismo carácter de texto plano.

De esta forma, se van sustituyendo los distintos caracteres homófonos por un mismo signo, y de esta forma se podrá hacer un análisis de frecuencias para descifrar el texto.

Pero si la repetición de palabras puede permitir el ataque a una sustitución homofónica, reemplazemos las palabras cuya repetición es inevitable por un nuevo signo.

Esta idea surgió a finales del siglo XIV. Las claves para cifrar eran sustituciones simples a las que se le añadía una lista de palabras con sus correspondientes signos para cifrarlos, y donde se incluían también otros caracteres, llamados *nulos* que no significaban nada, pero dificultaban el criptoanálisis. Surgió así lo que se conoce como *nomenclátor*.

Un nomenclátor es entonces un catálogo de sustituciones, donde además de los signos que sustituyen a las letras, aparecen otros que, o bien son nulos, o bien sustituyen a bigramas (grupos de dos letras), trigramas, palabras completas e incluso grupos de palabras. Pueden también incluir o no caracteres homófonos.

Obviamente, la seguridad de un nomenclátor depende de la cantidad de signos que se empleen. Si contiene un número elevado de signos homófonos para las letras, bigramas y trigramas más frecuentes, si incluye las palabras más frecuentes, y si además consta de un buen número de signos nulos el criptoanálisis de un criptograma resulta bastante complicado. Ello lleva, entonces a construir nomenclátores con miles de sustituciones. Tendremos entonces un libro que habrá que mantener en secreto, lo cual también entraña sus dificultades.

En España, durante el reinado de Felipe II es cuando más se ha utilizado la criptografía. Comienza de forma esporádica a mitad del siglo XV en la corona de Aragón, y se va extendiendo durante el reinado de los Reyes Católicos. Cuando España se convierte en un imperio se convierte en imprescindible. En un principio, se usaban sustituciones del texto llano por números romanos, pero eso resultaba difícil de entender incluso para el legítimo receptor, así que poco a poco se fue abandonando y se fue implantando el uso del nomenclátor.

Felipe II usó diversos nomenclátores. Éstos contienen ya un número importante de sustitu-

ciones y así como signos homófonos. Pero hubo descuidos en su diseño, especialmente a la hora de cifrar bigramas, que facilitaban su criptianálisis.
Aún así, no era fácil descifrarlos. Uno que lo consiguió fue el francés François Viète, con un texto que empleó el rey para comunicarse con el duque de Parma, que comandaba las tropas españolas contra el rey francés Enrique IV. Dicen que Felipe II, que consideraba sus cifras indescifrables, pensó que Viète debía haber empleado la brujería, por lo que solilcitó al Papa su excomunión. El Papa no accedió a tal petición, no por no creer en la brujería, sino porque ya sabía que los cifrados españoles eran vulnerables.
Un error cometido a la hora de confeccionar criptosistemas es que los que se dedicaban a ello eran independientes de las personas que se dedicaban al criptoanálisis. Si quien elabora un criptosistema nunca ha roto ninguno, no conocerá las posibles debilidades que pueda tener, y luego serán aprovechadas por los criptoanalistas. Esta tendencia la cambió el cardenal Richelieu, que tomó al joven Antoine Rossignol para resolver criptogramas interceptados, así como para elaborar las cifras francesas.
Antoine Rossignol, y su hijo Bonaventure Rossignol inventaron un sistema de cifrado conocido como la *Gran Cifra*. Luis XIV estaba tan impresionado de su trabajo, que les asignó un papel importante en el desarrollo de la política diplomática francesa.
De hecho, la palabra rossignol se convirtió en argot francés en un artificio que abre cerraduras, en paralelismo con su habilidad para "abrir" cifras.
A la muerte del padre y el hijo, la *Gran Cifra* cayó en desuso, y sus detalles se perdieron. Así que muchos documentos cifrados dejaron de poder ser leídos.
Uno de los mensajes cifrados hacía referencia a uno de los personajes de la Historia de Francia: el Hombre de la Máscara de Hierro.
Durante dos siglos, este mensaje permaneció sin ser descifrado, y fueron múltiples las historias y leyendas que surgieron en torno a este personaje. La más popular era que se trataba de un hermano gemelo de Luis XIV, condenado al encarcelamiento para evitar controversias sobre quién era el heredero al trono.
A finales del siglo XIX, estos mensajes llegaron a un experto del Departamento Criptográfico del Ejército francés, Étienne Bazeries, y dedicó tres años de su vida a su desciframiento. Finalmente, descifró una carta escrita por François de Louvois (ministro de la Guerra de Luis XIV), en la que enumeraba los delitos de un tal Vivien de Bulonde, comandante responsable de dirigir un ataque a la ciudad de Cuneo. Según el ministro una de sus acciones (una huida en la que dejó sus municiones, y abandonó a muchos soldados heridos) pusieron en peligro la campaña, y el rey consideraba entonces dichas acciones como un acto de extrema cobardía.

> Su Majestad conoce mejor que nadie las consecuencias de este acto, y también es consciente de lo profundamente que nuestra fallida tentativa de tomar la plaza perjudicará nuestra causa, un fracaso que hay que reparar durante el invierno. Su Majestad desea que arrestéis inmediatamente al general Bulonde y hagáis que sea conducido a la fortaleza de Pingerole, donde lo encerrarán en una celda guardada por la noche, permitiéndosele caminar por la almena durante el día cubierto con una máscara.

Aún con esta carta, hubo quienes argumentaron que Luis XIV quería encarcelar realmente a su hermano, y que fue dejando pistas falsas, como esta carta. Así, Bazeries cayó en una trampa tendida dos siglos antes.

Sin embargo, tampoco está claro que la carta dijera exactamente eso. Bazeries descifró el contenido de la carta, que terminaba *pasear por las almenas con una 330 309*. Los números 330 y 309 no habían aparecido antes en ningún lugar de los códigos cifrados, y por tanto no se sabía como descifrarlos. Parece ser que 309 representaba un punto, pero no había ninguna pista sobre 330. Fue Bazeries, quien sabiendo que el Hombre de la Máscara había estado en Pignerole supuso que 330 se correspondía con la palabra *máscara*

Como vemos, la criptografía iba tomando importancia y su uso se hacía cada vez más frecuente. Entonces, para el siglo XVIII los gobiernos se iban haciendo con equipos de criptoanálisis que trabajaban juntos para descifrar las diferentes cifras monoalfabéticas que llegaban a su poder. Estos centros eran denominados *Cámaras negras*. La primera en crearse fue la Cabinet Noir francesa, pero la más eficiente era el Geheime Kabinets-Kanzlei de Viena. Las cartas que debían ser entregadas en las embajadas que había en Viena eran enviadas primero a la Cámara Negra, a la que llegaban a las 7 de la mañana. Allí fundían los sellos de lacre, y un equipo trabajaba en hacer copias de las cartas. En menos de 3 horas las cartas habían sido selladas de nuevo y devueltas a la Oficina Central de Correos. Cada día unas 100 cartas se filtraban por esta cámara negra.

Las copias pasaban a los criptoanalistas. Estos eran funcionarios, bien pagados, que se elegían entre jóvenes con conocimientos en matemáticas y en alguna lengua extranjera. Entonces se les entrenaba para el criptoanálisis, y si superaban las pruebas eran enviados a algún país extranjero por un tiempo para perfeccionar el conocimiento de la lengua.

Uno de los mayores éxitos fue la ruptura de las cifras de Napoleón.

Las cámaras negras estaban volviendo inseguros todos los cifrados monoalfabéticos y nomenclátores, así que los criptógrafos tuvieron que adoptar otro sistema de cifras: el cifrado polialfabético, que había empezado a ser descrito durante el Siglo XV, aunque el más famoso es el conocido *Cifrado de Vigenère*, denominado por él como *le chiffre indéchiffrable*. Este tipo de cifrado no se había usado debido a su dificultad.

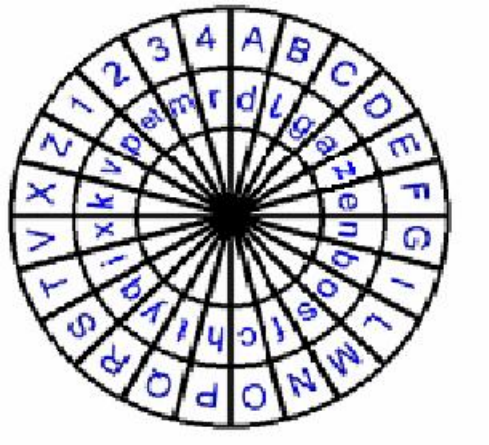
La aparición del telégrafo parece ser que fue la causa de que las cámaras negras cerraran su actividad a mitad del siglo XIX.

2 Cifrados polialfabéticos

En los sistemas de cifrado vistos en la sección anterior, en principio cada carácter era reemplazado siempre por el mismo símbolo a la hora de cifrarlo. Luego con la introducción de signos homófonos, y del nomenclator la situación varió algo, pero un signo de un criptograma venía siempre de, bien una letra, un bigrama, una palabra, pero siempre de la misma.
En los cifrados polialfabéticos, un signo de un criptograma puede venir, según la posición que ocupe de dos o más signos diferentes del texto plano.
Veamos un ejemplo muy sencillo y luego ya explicaremos brevemente la historia de estos cifrados y nos centraremos en el Criptosistema de Vigenère.
Por ejemplo, supongamos que queremos cifrar el texto EL ABECEDARIO LATINO
Para ello, las letras que ocupan una posición par las vamos a sustituir por la que le sigue en el alfabeto, mientras que las que están en posición par las reemplazamos por la que va dos posiciones más en el alfabeto. Tenemos entonces:

E	L	A	B	E	C	E	D	A	R	I	O	L	A	T	I	N	O
F	N	B	D	F	E	F	F	B	T	J	Q	M	C	U	K	Ñ	Q

Vemos como la A, en alguna ocasión se ha cifrado como B, mientras que en otra como C. Por otra parte, en el criptograma vemos la F, que en un caso procede de cifrar una E y en otro procede de cifrar una D.
El primer cifrado polialfabético lo dio a conocer el italiano León Batista Alberti, en el Siglo XV. Este sistema consiste en dos discos móviles, concéntricos, unidos por el centro. El mayor fijo, y el más pequeño móvil, y cada uno dividio en 24 casillas.



En el mayor escribe el alfabeto en claro, ordenado, omitiendo las letras H, K e Y del alfabeto

latino, y los cuatro números 1,2,3,4. En el interior van las 23 letras del alfabeto latino, en orden aleatorio, y la palabra et.

Emisor y receptor deben tener discos idénticos. Eligen entonces una letra del disco móvil, por ejemplo, la n. Para cifrar, el emisor elige una letra del disco grande, la escribe en mayúsculas al principio del texto cifrado y la hace coincidir con la letra elegida previamente por ambos.

A continuación cifra unas cuantas palabras según la correspondencia determinada por la posición de los discos. Repite la operación: elige otra letra del disco grande, la escribe en mayúscula en el criptograma y la hace coincidir con la n, y cifra otras cuantas palabras según la nueva posición relativa que tienen los discos.

El cifrado así obtenido es polialfabético, porque los sucesivos giros de los discos dan lugar a distintas sustituciones, y por tanto a distintos alfabetos de cifrado.

Además Alberti incluye una tabla con números de 2, 3 ó 4 cifras, empleando sólo los dígitos 1,2,3,4; y al lado de cada uno, una palabra o grupo de palabras.

Entonces, estas palabras o expresiones se reemplazaban por los números correspondientes, y se cifran de la misma manera que hemos explicado previamente. Por ejemplo, si el número 123 representa a la palabra *siempre*, entonces, según la posición del dibujo se cifraría como *petm*

Este método de cifrado es mucho más seguro que los nomenclátores empleados en su época, y es además el primer cifrado polialfabético de que se tiene constancia.

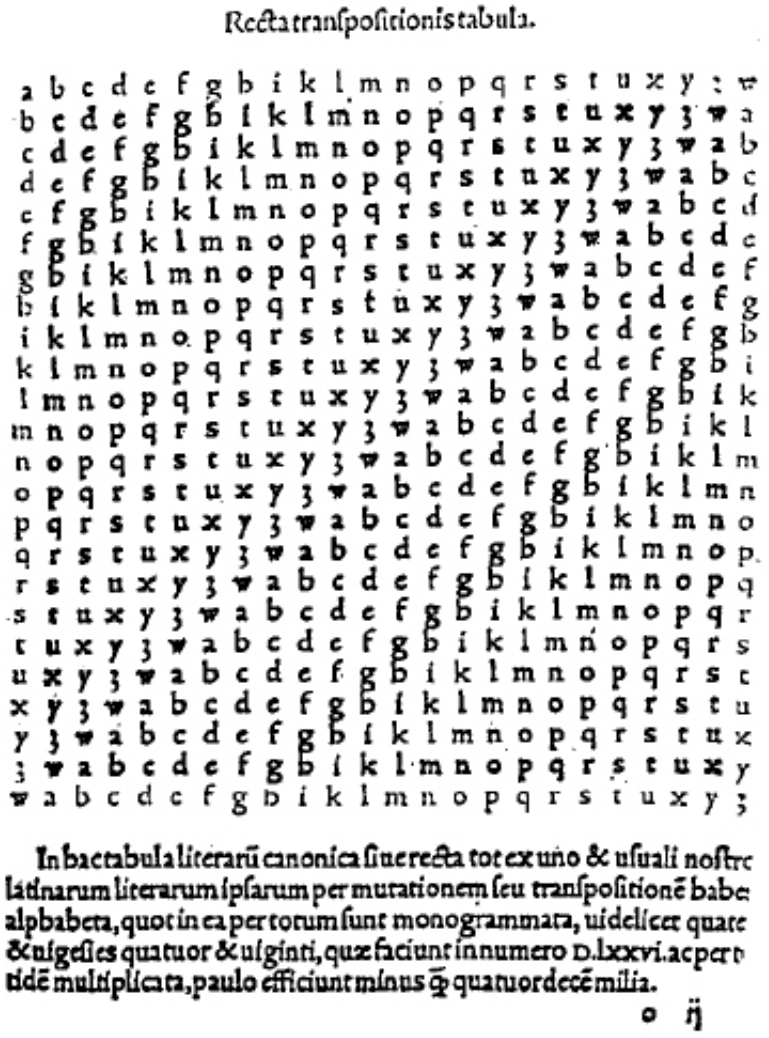
El monje alemán Trithemius concibió otro sistema de cifrado polialfabético. Utiliza para ello la tabla siguiente

Si se quiere cifrar un mensaje, la primera letra se hace con la primera fila, la segunda letra se cifra con la segunda fila y así sucesivamente. Por ejemplo, *criptografía* se transformaría en *cslsztnwiptm*

Si bien este criptosistema es muy débil, la tabla de Trithemius ha inspirado a otros criptografos que la han utilizado para describir métodos de cifrado. Por ejemplo, consideramos el caso de Belaso, escrito en 1553, en el que utiliza la tabla junto con una palabra clave. Por ejemplo, si la palabra clave es CRIPTOGRAFIA, entonces se utiliza la fila que empieza por C para cifrar la primera letra, la cifra que empieza por R para cifrar la segunda letra, y así sucesivamente. Por ejemplo:

Clave:	CRIPTOGRAFIACRIPTOGRAFIACRIPTOGRAFIA
Texto llano:	NOPORMUCHOMADRUGARAMANECEMASTEMPRANO
Texto cifrado:	PFZDLABTHTUAFIDXTFGDASNCGDIBNSSGRFXO

En la obra *De furtivis literatum notis*, escrita en 1563 por el napolitano Giovani Battista Porta, se recoge toda la criptografía hecha hasta entonces, además de varias aportaciones del autor. Uno de ellos se escribe usando el alfabeto latino del que se ha suprimido la letra K. Porta construye una tabla con once alfabetos



Recta transpositionis tabula.

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

In hac tabula literarū canonica siue recta tot ex uno & usuali nostrc latinarum literarum ipsarum per mutationem seu transpositionē babe alphabeta, quot in ea per totum sunt monogrammata, uidelicet quate & uigesies quatuor & uiginti, quæ faciunt in numero D.lxxvi. ac per tidē multiplicata, paulo efficiunt minus q̄ quatuordecē milia.

o ij

AB	abcdefghilm nopqrstvxyz	OP	abcdefghilm stvxyznopqr
CD	abcdefghilm znoqprstvwy	QR	abcdefghilm rstvxyznopq
EF	abcdefghilm yznopqrstvx	ST	abcdefghilm qrstvxyznop
GH	abcdefghilm xyznopqrstv	VX	abcdefghilm pqrstvxyzno
IL	abcdefghilm vxyznopqrst	YZ	abcdefghilm opqrstvwyzn
MN	abcdefghilm tvxyznopqrs		

Para cifrar, al igual que en el cifrado de Belaso, se toma una clave (por ejemplo, CIFRADO), y cada letra del texto llano se localiza en el alfabeto determinado por la correspondiente letra

de la clave y se sustituye por la que está en la misma columna. Por ejemplo:

Clave:	CIFRADOCIFRADOCIFRAD
Texto llano:	UNIVERSIDADDEGRANADA
Texto cifrado:	IETDRFAVZYVQQNFVCRQZ

Pero el autor del Siglo XVI que ha pasado realmente a la historia es Blaise de Vigenère, un diplomático francés, que a la edad de 47 años abandonó esa ocupación y se dedicó a sus estudios. Entre sus obras, nos interesa la denominada *Traicté des chiffres*, publicado en 1585, y en el cual además de lo que indica el título trata temas como la magia, o las ciencias ocultas, lo que hace pensar que para Vigenère, la Criptografía forma parte de lo esotérico.

En su tratado hace mención a muchos de los cifrados que hemos comentado aquí, así como a otros, y entre ellos aparece uno que él denomina como *le chiffre indéchiffrable*.

Este sistema, mucho mejor que los que existían en la época, ha permanecido, no obstante, oculto durante más de dos siglos. Fue en el siglo XIX, cuando los nomenclátores empezaron a caer en desuso, y con la aparición del telégrafo y transmisión en Morse de los mensajes cuando hubo que cambiar el método de cifrado. Sin embargo, no se tomó el criptosistema de Vigenère tal y como él lo había descrito, sino que se tomó una simplificación que lo convierte en equivalente al cifrado de Belaso.

Al igual que Belado, Vigenère utiliza un cuadrado con tantos alfabetos para cifrar como signos disponía el idioma, pero a diferencia del de Belaso, el orden de cada alfabeto no tiene porqué ser el natural.

Sin embargo, la mayoría de los textos acutales identifican el cifrado de Vigenère con el propuesto por Belaso.

Vamos a continuación a analizar con más detalle este criptosistema, así como las formas de criptoanálisis, debidas a Kasiski.

2.1 *Criptosistema de Vigenére*

Como hemos dicho, actualmente el cifrado de Vigenère que se estudia corresponde realmente al de Belaso. Para nuestra lengua necesitamos una tabla 27×27, donde en cada fila (y en cada columna) nos encontramos las 27 letras del alfabeto. En la primera fila en su orden natural, en la segunda comenzando por la B, y así sucesivamente.

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

Ahora se elige una clave, que consiste en una sucesión arbitraria de letras del alfabeto, sin tamaño predeterminado. Por ejemplo, tomamos como clave IGNACIO.

Si tenemos un mensaje a cifrar, repetimos la clave tantas veces como sea necesario hasta obtener un texto del mismo tamaño que el mensaje. Tenemos así una correspondencia entre letras de la clave y letras del texto plano. Cada letra del texto plano se cifra buscando en la columna que empieza por dicha letra y en la fila que comienza por la letra clave correspondiente.

De esta forma, obtenemos el mensaje cifrado.

Por ejemplo, el texto

ENMARZOVOLVIERONLOSGITANOSESTAVEZLLEVABA
NUNCATALEJOYUNALUPADELTAMAÑODEUNTAMBOR

quedaría cifrado como sigue (donde hemos separado de 5 en 5 para hacerlo más legible)

MSYAT HDDUX VKMGW SXOUÑ WBGZO UMHBG IEBSZ MBNBC
UJUIN TCSSQ ULUOI ZCVND GSIIR NÑQLS CSGAÑ JDZ

Si tratamos de ver como trabaja este criptosistema, vía la correspondencia entre las letras del alfabeto, y los elementos de $\mathbb{Z}_{27}$, en realidad, lo que tenemos es que la clave es un elemento $k = (k_1, k_2, \cdots, k_t) \in \mathbb{Z}_{27}^t$, donde t puede tomar cualquier valor (en este caso, $t = 7$, y la clave $k = (8, 6, 13, 0, 2, 8, 15)$), y si

$$m = m_1\ m_2\ m_3 \ldots m_n$$

es un mensaje, su correspondiente cifrado es

$$E_k(m) = m_1 + k_1\ m_2 + k_2\ \ldots\ m_t + k_t\ m_{t+1} + k_1\ m_{t+2} + k_2\ \ldots m_n + k_{n \text{ mód } t}$$

Obviamente, la función de descifrado, D_k lo único que hace es restar $(k_1, k_2, \cdots k_t)$
Para criptoanalizar este sistema criptográfico disponemos, en principio, de dos métodos.
Primero (Método de Kasiski).
Aunque el método aparece descrito en el libro *Die Geheimschriften und die Dechiffrirkunst (La escritura secreta y el arte del desciframiento)* escrito en 1863 por el prusiano Friedrich Wilhelm Kasiski, parece ser que el primero en utilizarlo fue un tal Charles Babbage, en 1854, un científico británico algo excéntrico.
Fue él, por ejemplo, el primero en observar que la anchura del anillo de un árbol dependía del tiempo que había hecho ese año, lo cual permitía, estudiando árboles muy antiguos, determinar climas pasados.
Debemos también a Babbage el que el precio para enviar una carta de un lugar a otro de un país no dependa de la distancia entre origen y destino. Y es que él observó que el coste del trabajo necesario para calcular el precio de cada carta, en función de la distancia que tenía que viajar era superior al coste del franqueo, así que se adoptó el sistema de precio único.
Diseñó una máquina calculadora que podría ser programable, y que podría ser la precursora de los actuales ordenadores, pero no llegó a completarlo debido a falta de financiación. En el diseño se incluía una *reserva* (memoria) y un *molino* (procesador), que le permitirían tomar decisiones y repetir instrucciones, tal y como hoy hacemos con los comandos "IF...THEN" o con los bucles.
El desciframiento del criptosistema de Vigenère por parte de Babbage pudo proporcionar a los ingleses una ventaja decisiva en la guerra de Crimea, pues al parecer el Zar Nicolás I empleó un cifrado de Vigenère. Por tal motivo, parece ser que la Inteligencia británica obligara a Babbage a mantener en secreto su descubrimiento, lo que proporcionaba a los ingleses una ventaja, que duró 9 años, sobre el resto del mundo.
La idea de Babbage y Kasiski consiste se basa en que si encontramos grupos de palabras (bigramas, trigramas, tetragramas, etc) que se repiten en el texto, si el número de repeticiones es mayor que la longitud de la clave, entonces sus cifrados correspondientes aparecerán también repetidos (pues el número de maneras distintas en que pueden cifrarse coincide con la longitud de la clave). Además, si encontramos varias repeticiones que provienen del mismo grupo de letras, el número de caracteres que las separan debe ser múltiplo de la clave. De esta forma, podemos estimar el tamaño de la clave.
Vamos a ver un ejemplo, y todo quedará más claro.

Consideremos el siguiente criptograma

UOFSL GÑAPW ZPGEZ RSOCU QSCOP LTDWP DEYOD SXPES XRPDB
GBJPS QKOGT AVOWO EEZZL OWDQO RONIN RSHBU QHEDO EGOVA
ADBZL EASUQ SDYDQ ZLHÑA FODFG EPSDR PTBRL HBOPW LZWES
OCHPA XGSDK AÑODI PCQGD SOENU FOÑOZ WQJML QHPJP RMDJI
DAMDV OECNZ VSECB BEOYB HSXHP NGWÑD CUQBS BRUZO NOEAE
SNWNI NALHE OXFFS ATEOL ZMHBG LRPLO OVDDE ZEZQA SNCZH
XAODX RAFHS FBMAX ROOXA FDCRP NNRLN WAÑDC WASNF FSÑUN
ZBJTE EOÑSW AFQZB ACURL HSAFI LSYTB BNSEP BGDJE HNPSI MNGSD
SDAQB GSDDN RFBMA XROOQ EXWKR ANPSX OOIQS COXAL DCRPT
ESSBF ANCZH KDBBÑ SYAPW OVMBU OWJPR GDÑSE DQZZH FIQAA
DEDQZ LTGNP ONWAN VDDSM ROOÑW ABHSX RTAOD XHFRH NZIDA
YELHK JNJVO EEZEZ QATUS WEALX SXDOE GJCET AXSDQ MNNGS
DEAMJ VSUOF NASFI EGZXA SZDDD WOXOA GAPUO NOEAF WXDFO
PODZM SPSVO MLPSL SWCBB NWPRG DÑSFA ZIZHB AVOCD EDUHE
WYTBH VZPGB ODSDT NBLIG RPWÑD DQHSF GEUXO DSFAC DVDEO
URZHÑ OZQOG MDQOM SUAFE LGMNB EOGOE ESVHP NGWÑD OEXOC
SMLUR LRWAC GSAPR NKOÑC UQZVS ROXOE GTBHR OAPLD JSOOE
FKOBO IQBÑD NOXOD RPVUR CWAPN GLSWD BZZGO EOOMS LAGDÑ
DPLYJ XRASQ HZGBR QBÑWA DQFFS SUÑWO GMNCD ÑWOOQ BNDYT
EOCOC UQZVO MLPSL EPRPW ÑOPNQ ZDDBO EROZM CUSXO RALZZ
HRIGO XDECB BPSEA EDXFG EFSRO NINBZ GTEZI LRAPB GOZÑA ZIZRP
LBHAO UAEDD OCUQZ OHBIE WEJOE UBSQT AGWGO EOOWL ZOEFO
AODEO WZSYP BQZIT EYEZO DRNHE GMDBE ZGWAR WOPDE PSVDE
IYOXS ELBHN OWCHZ ZHMSG GZBAM UQZHW OFHFS ZOFRO IDAFA
FIMCU DXNWA FOXHT AFROQ ANBQO GWAFA LGMVU ZVOED QZWJY
DBROS XPESX RPDBG JZTMC WZXAS QOCQM DUDMJ PNPWL HPCBB
GWDTU DOBGN TDWPD EPSLH BEOIZ VALSO KOYDQ HNJTD NRZSY
EXKOH FIEQZ BGNNP LGNAF OVKMJ QFFSG RFJVO WOSGL PMCHO
ÑGMRN RFGMS CSXOE CBBFB ÑUOVS ZWOPS NDÑIZ OXDQA XIZFG
IQBVD ÑOZHS RPRNG LKTCG WWOOE NZQJY EKICO ZOFDC ITLQU
SDBEE DROET NZZHX AFQZB HEZQS RASPS DJWOO JCOMB NBÑDY
AEDXI DAÑOT DKFNA SZTAF ELGMS QUFWD LBQFO YDBHO SÑHBO
VVAMÑ GZHGS TSCGM MUSXI MSPSÑ SEMBB EODYC WÑWAE XQZBÑ
UEHZR PTBRZ HBAEO LPDIE JXOFR BQROC UQEFH TEEOL AMCBB
ÑDPNO DXIMC GDNDY LBHQG MNPSD WYVQB EDEJB HOODC NRSDN
UQBÑW MISBZ GMBNE ZGÑOY EVSFO XOQSA GEOPW MDQZL GPGUD
XHMBU OBJPH NQSOP LBGSS YTQSD IMBNZ LHTEE GLWXP QBOID
AÑZON MLBIC DWAPD ÑSWAF WOGDA XOLBF ISJLQ TUPOÑ RPRUD
ROÑHN RZBOE QBOEA CNHAO EAPOD HPGHB VSSAÑ WLQAN GOÑDP
LCGSA PRNJC SWINB ZPGEZ RSOEU NPFSW OFWCT DAZQS HORNY
OHPDN PLOWD QEZGF EPSNO LAEQL WXAZS DOÑAA DXOLO FFFSW

UQUZV MCUOC SXEZR LGKRQ ZVSYA EROEM JNELG MLXSG ODSQZ
ZHMLN GOWYA UHLPP LQBDJ UUISX IGDQZ JHGST DWPDE FQZBX
UVSCS EYZWY DEYNB SAMLQ HJIAD NQVOE EPSOB EEESD RAMQH
EWÑOF OEGMV QHLGA NXODW PREOM JECNB ÑDGNN HLZTD NO-
VAM RLOVQ MBBRO KPIZI SHPIF AOHPS PSDWE TUSCD YDQZL SXPES
DOKFH BÑODO ZOWOÑ OZRZE MRNBZ IPNQG BJPEY ECSYD QGOZÑ
AYWXD OEESQ GPSBS COBUQ HFBMR HILFG EZDVS TNGSC SEAÑO
ADDQH SDDWO CDÑWM CBBÑJ ÑIEZZ OWPNH LRAAX HFGPS GOMOY
LBHAO YTNBZ HÑUÑW OGFOF ROJYA QIOGY AZOEO HESSE OWYQZ
GOETB JXWHE EHZRP LNQSS YASOQ GMNPS BJPSQ UFBFE FISAA NUDÑS
WOFUS IMNBH NODEO WLRPL UASIP SXONW PNNUL UDAZR OHPCB
BPJYD UOLZA COWÑS YTQQZ BGNNS IIPNF WZBMC HOEWÑ AFWXV
ARUÑZ BFEFR ZBOET OMWMC QILQP OFROE TEXRO ZTCNR LQANO
OMSLA LIZGE OPSWJ UEEFF SBEER SOYAX DDBMV QULBF EFQZB
PLTSN VTZBR OHGSG SEOED QHNDX UZOVS ELBHQ WFAZD DBMVQ
ULPMN FSSHX EFSDE ARQHL GGTNO XIPSP SLZÑA ZÑLGP LOWXI GRB-
BÑ SFIQG COQIE AOEAR PDXRP PNHLP MNXOD AGLNH ÑSWCB GCSAD
QONJP RPDND YLBHN OWCHZ ZHOEV DDSMR OOÑWA BHSXR TAX-
OF BTCNE ZHTBU ZSRMD PSNDY TNQED ÑOZZL QTVUZ SÑMCU DXS-
DA XOCJF APSVB ARGSÑ SXOPD BJPDB IZRPH QGCOX IQBEO EDQRO
HXOZI ONMRY ODRPC NQOGT ANZZH XIFAZ HSOYP CSEQH SVDMC
BAAOZ AEDXS YLNTF BOAOW ZBOEY ONDYD BSNVA EZJXO XOOVS
ZMSHH SBETE JWSYT BHÑSA RUSXI MCUDX NEUFA LEMSL SWEDE
ZRSDW AGSWS DAEWL OHEZI FGMLB HAGTM QGZHO INHXD PNODX
IDAED XJYOÑ HEOÑU XDLED EOWLP WEPSD QPNPW OGANC DCZMP
QRCSR OFOCW NEEOÑ SWRUD ROETN SVZGG NGOBC UQOYD EAZIO
HSAÑW LBPNO DXIDA PDVOM RYOÑJ DAPSV UGEEG OGAYN ZVWBE
ZSEGM RBBLZ NOFFF SBOEJ XHPNP SCDOE ZOCOY JBHDW WVQHE
GPSNZ ESDMU BZRPL NECWX EEODS XAZOW OFAED XNMSN GZBGN
ISXOO OCSCD EEODX TARYO CDYCB BNDXE EZLAT TNRJH MLNGO
ZDEFI ZEMRN ZZHBR BMSAA SPWLH FRNIL PMNPS LEWAM OCQAN
QHLED EOOFQ TOZZL BPCQH SRMDP SDSRU UGNDX IQBÑD RUNQL
AMYNH NJKAO OCBPA MJVIP NUOFB MSCSC DEAÑD CRPAX ASÑÑL
QZFSR OPJCO YTQAL HOEPW OÑOIN HXDHO XKSSD OZOGS DEXHZ
ZPLFJ OZASQ KZZHI BPVOY DBNRJ XEPDN DXOOS XWLAI DVQMN
UQLNW AISQS FAOWZ BQUQQ LRMVQ ÑWOEI ZHSRT OFOJH PHUQS
SDOZQ LRMVQ ÑWOEL QXLBA SXDDU DIGDD RPLBH AOUAE DDNWA
ÑJVZM RNBQO OEXDD AANBH JSWMH BÑDEE IDVKT OGGSH FECOC
OEIQA AGPLB HRDXB ESDRP LNSIE PDUQS DYSQH SBFIQ GZBMB EJ-
WOO OFEZG EUFGO QGEER ZHXAF OXITG HDDSY ADJOZ BAEOS HADQV
FAPDN RJHTL QBNWA AZIOG TOEOV EPCNR ZDDIS WXOWD BBÑSW
AFPZI MSFSR JYDUO XSYPB ÑZHOE NQOWF EFVFA PAZIO HKLBH WOÑHQ

IOHOE FICDL AÑOXZ TRUDD HMNSG SSYTB HJHML NALBO RNHÑD
DAPOD RGRNB ESGNN HOAMN NQLHT SUBRO NLNGL KMNMO CDYCB
AZHAN NAMJW OFEZG GNHBS KPRFD ÑSBEF OÑJXB ESLZG MÑGLR
ASNEO BMSCD CJYAG SXJPR QKOGN EEONW ANPSS BEEOI ZHWUY
WXDEO FNNDY LBHAJ WMBBO HMGBP SOOOF EZGGN FDPDÑ AZIOD
WOERO HMNSG OBAPB RSOYR QUCSE AEEZG CUQZL IDOOV LFGEU
PLBMB EWOBO ONHFE MSBHO KALIW LOÑEE GLGPN CDNDF IQAAD
ÑOZJX OHESS EOÑIB BXJPV NFFSÑ AFWGS TAZQC SÑEEO XIPSH HZXAS
ZDSAB OEILR PCUOT DEENG NOOIB PFSYD UOVDP SQBNW MLQHX
DBEER OGWAB GSSYT NQSDY SUSWE DECSX RTEZI ORPLN PCJUU
XODWR UUDQJ TAZRZ OEUFV ZANRQ HROÑI NSVBA RGSSB HIFWM
ZPHNH EOCUQ ZZUDA EDXHM LUGÑS WAESQ WANQB NOYTN RLSDA
HBLBA CTSÑS YSNHS BPSGG OZWAF EOGAL NDDQG RURLR PSGOM
OTMCG OUYAP OADDU ZOSGP NHSGD KLUAA WAASD EOOOF EZGWA
CGZZA NSOÑO FRNKO HTAOD VUMRB BVOEH NALQM SLRFG XIQGZ
BMFBB ÑDBOE ECWXE EOGSL EZRZH EEYOX OECHO XRADQ HAS-
DT NGZBK AODXS WSBZL ZFOFS BJPDN GZBBA FALRA SPSPO ECUBL
QTOZT CSYTQ OOZWO FGZRP APDÑS SEXSN VASLE LZXEE ODPWA
ZQZNB OXKZG TEZIZ SYLNH SZPNO WZHML HÑÑSW AYOYO YAQHE
ONAHB OBARY SQOWE BBOHB AADVZ TGQGL APNGS GDWTQ OÑDME
FICWN OEROH GAEPZ ZMDHG LWYTN QEOÑO XULPM NXODE TLGGL
TMSQH NJMLU RLHOE XKOZM MQBOB FRQXL GÑINH LRARZ OÑOED
QDCFG IPSLH PLOOD QACHP SSDTB QZBGN NIOGE AODCO LAPSC SX-
OEO ASFRU TSQMD NNWJE GBISS DNBSD IMBNT SGXEY SXIPE ZQVOH
APDOB GNFJO ZADQE SSORN HEDOA XOOHF RHQEJ DACOC SÑIND
NJBAE JXOXB UIZED OCWZJ YEFEL QTOPS DDWEP OÑNOE BZGWO
OISÑO OONZZ HHIOW ZHOEX ISSXP BNLZM SODDI GMÑGO HOEXD
DEMJN GZHPN QZSBF EEWZG CUQZZ HPXCS ÑWÑIB BLGTO FSIEW
OEOCD YCBBF BQEEK ZGEIS WVDEO ZDRON INBLR MMNHB JPUZO
AGPTN RZPAS DJORP FXDCS EEXVL ZWAMU ZRPLS OVSAN UBÑWÑ
IBROZ MPEDI WXIPO ÑRPLY OCFGE ÑGLBF OQZSA BEGJÑ SUOFS LGÑAP
WZPGE ZRSOÑ OZHSR PRNPL QAMBJ XONUE ZLRPS HICOH IQHZR PSG-
WX DSAÑS CPGSO OÑDPL YOCHT NQBND YTEOC ZAAXE CSÑIB ROHMC
EWPWÑ IBHJE PNNZS RMDQH DWYCH SXIAY TOMSD LBSXQ ANGGL
RAEZI ZBÑEF HSBNU FQLGW ONICO HEFOÑ DPNFJ NOXIZ DNDXO HBZPE
TNQFZ AIZHL ZHAÑZ OAGCT DDOZO FROHB UQHOZ ÑOEDX SWAHG
OZTAZ DMJPN PWLKA LIWZO MTEOG SEAEO CSRIB BNJMN PDOGM
YNJXO DUGOC SRUXO CRPLO DCGPO LZZJY IODBJ PEZQZ BFRBR OZMNN
KOTGE QZNDE TUZVO DCNGM DYIMO ÑDPNY SÑWAD QJXQM MCDÑS
MMNEZ ZMSFD VDPNG DXQPS ODXKP NOWÑD OEDJO OCUQZ VOSIF
IZGTA ZDRON INHSR AUZSX UPNPG ZRPLN WWORI ZONWA NPSDJ BAP-
GO HPPES QJYTB QZAAH NPSOB OPWÑD PLSOV SANNR OBFRN GDSSA

FILSE ECJXI AEZIS SDRNT SGXEC SCDUO FSLGÑ APWZP GEZRS OY-
OFS AZMNG SZSEA UBBJT EGJÑQ GAZRZ SYCBB EGAEX ALGML OOM-
DO EBICD ECHOE GADUO DRPVU OTSMD BQOYT LBAOI DOFRO RTS-
GO XQTAP SVUML QDXHG SFJOC ASGSC ATNNP LBQRQ BESPS QAL-
GÑ OXDCR PCQBS ÑMEFE FAASB NDJÑI BFFSY OYSCS ÑINZZ HDIQH
QDEYF ONGTF UQSDE DQHFO HEZIF GMCNG LXAGE WEDXA ODXRA
EFILG ADQOÑ DOENU FOBOE IZRMS COCIP SXOSR PAPSF BXAOD XRAPQ
BSBEU XOCED EIOVS ÑIBRF GMNGS WJÑHB ISSXP BWXHB IEOÑO PN-
QZW OBANG MWFRN GSDCU QRSPG JBXZH PAEQL RTOÑJ OBOIN OVGPG
ESDDO EFJOM BEPWN WANXD EGMZB QZBDA ÑWLSJ ASSCO YDBRO
AMLNT OZMSP WPWÑU XILRP SPSND XUZWN OÑIBB NDXOC OCOÑA
FISUM RFSLH TMUHW DBOEZ LONSB ZFIMF NZEOO EFSXI TDBQZ BCUQS
VWRIB SVZGG NGXJY CNZVS RAESW DEAZW XUGNN ELGFE FSVOX
EZILP MAZIO JDSHZ LOCUU BZHSE YDDRP PHRCW DEZKS RMSUB CSÑIÑ
WCZAS ÑSXSQ IOWZH OEXON WPNOW LSEAO SCITD HAMGP RHA-
SO OAIOC WASYS DSEEZ SVQGA EISIA DQZVO NOEOE DDIBZ ZZWEI
DLQAN OSMWD EXECD KEOIZ RPTEO DZMDN GLAMC BBÑDM UZZFU
MRYOD EDOCW NWAPQ GZSET NKOÑG RFJVO EENBE WÑICD LHGSP
SDWRN UDDTP BEWVS EEZJX OEEOG OIMEU AAZMC NPVSW AÑDCR
PHBGW WRUUI LEDEP WDEGS BOVOE MHXOG PSPSV OMLPS LQANG
GLZMV QZOWO APROH GSTDW PDEFF FSKAQ AASLA ÑOXOB RQELG
MRFSA ODAXO WJOAZ ÑLXAS QOCQM DUDMJ PNPWL BASHE ZSYQH
SWDXE ZIZBT EZKSG FUPRO FGERJ OGLAF OÑKPR FODHG SCZLB
PSFSP JPRBB OBDEP OXRAE ZJXOX AEOYO OECGO IPXGD DQANG
GLITE YEZHK EIODW HAFVL HFAOD XKPRG WCHPE ZEFGM YFWWE
WEUZF HTOZJ CHGLN ZZDNS QGGDÑ OZJXO MTQBN WANUB ZQPNG
SJVMS GODWY TUDAD DEXJX EACBR OETEP OÑZMM NCLBM EZFFS
WOQBN DYTED OBPLO JLGFI GDÑSW FBBÑD ÑOYSX IMNPD OBFRQ
RSSYT QHDJE SHSYD EDQAF RMNMO WWPNG GLHÑO XDNON AQBDJ
ECNXL HARUU SBMLQ HVOEP USKOE DQZVO NOEOE DDIBZ ZRPJB
IOGXI ZOCZA DQXZQ WAIOC ZMSOO TOEYC DXSDS HHSBT CUOVS
EEZQS AMCBB FBSIF DADPN GWXIM DBHSB SAOSC ZPNUB QJYRQ
ECDÑH QEOGA SNPSS YDBNL FGEQZ DONIN EZGCU QHOZA OLDÑS
ÑIESX HGSFD CRASY DXDWO SDDFG EXDDV AMÑGO HOEXE FSNLB
BZZAS QQFBO AEWLB PNFJO ABRQH LHALB QFOYD BSWEP ZBOÑS
EMBBE ODLNE FSDTN ROZÑU NGEWF OHGDJ WAFSL IDEIW ZOBRQ
UFBFA EZOEA RDJOZ AHNQS OKEXZ OQANG SDIAC BBFBM CUSCI MAY-
OC UGRNE FSETB FFSYA PWOFG IQGOW DSQBZ HTRQA ZHEOX DDJDS
HZLBA SQOVI PRBBZ BASUG OAASP WTDMQ HWXDE QHSÑO XOFEZ
GCUQO BJTHQ AZHFE ZWÑDG NTWTD FOPOG WMNBI OBPMB HFBXU
QGEDO IVDOZ GNBBZ SEDQB SBRUZ OAODT QASSY TEODB ATQBQ
OGNYJ OGFOÑ OTDWA GWOGD AHGDJ WAESA ZTCBQ ZBGNN HFOHE

RWCAP ZNHSS ENQQO HMRUD BJPYB AOAGE EOAOD ADJOH PQHSÑ
SYADJ SAPMH SCDUO FSLGÑ APWZP GEZRS OYOOG ONAQH SPJPR
NILBD ISWÑO WAIDV JYTNR ÑSEUY JTSDT EOEDO EFSÑJ ÑIEZL QAN-
QZ RSÑHU ÑZRPS HTLBF AFWLQ ANXOA GAMQH LRPUZ AFBOO CGZRT
GUDDD OOZRO PMSGO MOÑOZ SNVMR HBZHW IDJSR ASYOQ WÑOFS
XZMTU SCGMP NGLFG EXODE WAZIL HOIQG LBQRH IZHMV BZFBF
APROZ SOYPC SKDBB ÑSEEI SXRTA ZOAGP CUDÑS NAEOE WWLBI
ZRMCX ODSOE NELGM TBHAO DAQZÑ DWOEE OGAUE HFZMF HSSBE
EZHSP WENHF QWAEW GWOEZ QSOPN ISKRP AZRLG BEZHL BOOQB
EJEAX DNOOA FBZKP LQGSO EDQPO HACHE LGFEP SEJEH UXZHD
ECZSQ AMUGL ZASOD WDPSG OXONA ZRZBM DBHLZ MBHSX OOEPW
ZHTGH OVFGE XDDPG REDD

que sabemos que ha sido cifrado usando un criptosistema de Vigenère.
Buscamos grupos de tres letras que se repiten, y encontramos, por ejemplo CUQ, que aparece 14 veces.

UOFSL GÑAPW ZPGEZ RSO**CU Q**SCOP LTDWP DEYOD SXPES XRPDB
GBJPS QKOGT AVOWO EEZZL OWDQO RONIN RSHBU QHEDO EGOVA
ADBZL EASUQ SDYDQ ZLHÑA FODFG EPSDR PTBRL HBOPW LZWES
OCHPA XGSDK AÑODI PCQGD SOENU FOÑOZ WQJML QHPJP RMDJI
DAMDV OECNZ VSECB BEOYB HSXHP NGWÑD **CUQ**BS BRUZO NOEAE
SNWNI NALHE OXFFS ATEOL ZMHBG LRPLO OVDDE ZEZQA SNCZH
XAODX RAFHS FBMAX ROOXA FDCRP NNRLN WAÑDC WASNF FSÑUN
ZBJTE EOÑSW AFQZB ACURL HSAFI LSYTB BNSEP BGDJE HNPSI MNGSD
SDAQB GSDDN RFBMA XROOQ EXWKR ANPSX OOIQS COXAL DCRPT
ESSBF ANCZH KDBBÑ SYAPW OVMBU OWJPR GDÑSE DQZZH FIQAA
DEDQZ LTGNP ONWAN VDDSM ROOÑW ABHSX RTAOD XHFRH NZIDA
YELHK JNJVO EEZEZ QATUS WEALX SXDOE GJCET AXSDQ MNNGS
DEAMJ VSUOF NASFI EGZXA SZDDD WOXOA GAPUO NOEAF WXDFO
PODZM SPSVO MLPSL SWCBB NWPRG DÑSFA ZIZHB AVOCD EDUHE
WYTBH VZPGB ODSDT NBLIG RPWÑD DQHSF GEUXO DSFAC DVDEO
URZHÑ OZQOG MDQOM SUAFE LGMNB EOGOE ESVHP NGWÑD OEXOC
SMLUR LRWAC GSAPR NKOÑ**C UQ**ZVS ROXOE GTBHR OAPLD JSOOE
FKOBO IQBÑD NOXOD RPVUR CWAPN GLSWD BZZGO EOOMS LAGDÑ
DPLYJ XRASQ HZGBR QBÑWA DQFFS SUÑWO GMNCD ÑWOOQ BNDYT
EOCO**C UQ**ZVO MLPSL EPRPW ÑOPNQ ZDDBO EROZM CUSXO RALZZ
HRIGO XDECB BPSEA EDXFG EFSRO NINBZ GTEZI LRAPB GOZÑA ZIZRP
LBHAO UAEDD O**CUQ**Z OHBIE WEJOE UBSQT AGWGO EOOWL ZOEFO
AODEO WZSYP BQZIT EYEZO DRNHE GMDBE ZGWAR WOPDE PSVDE
IYOXS ELBHN OWCHZ ZHMSG GZBAM UQZHW OFHFS ZOFRO IDAFA
FIMCU DXNWA FOXHT AFROQ ANBQO GWAFA LGMVU ZVOED QZWJY
DBROS XPESX RPDBG JZTMC WZXAS QOCQM DUDMJ PNPWL HPCBB

GWDTU DOBGN TDWPD EPSLH BEOIZ VALSO KOYDQ HNJTD NRZSY
EXKOH FIEQZ BGNNP LGNAF OVKMJ QFFSG RFJVO WOSGL PMCHO
ÑGMRN RFGMS CSXOE CBBFB ÑUOVS ZWOPS NDÑIZ OXDQA XIZFG
IQBVD ÑOZHS RPRNG LKTCG WWOOE NZQJY EKICO ZOFDC ITLQU
SDBEE DROET NZZHX AFQZB HEZQS RASPS DJWOO JCOMB NBÑDY
AEDXI DAÑOT DKFNA SZTAF ELGMS QUFWD LBQFO YDBHO SÑHBO
VVAMÑ GZHGS TSCGM MUSXI MSPSÑ SEMBB EODYC WÑWAE XQZBÑ
UEHZR PTBRZ HBAEO LPDIE JXOFR BQROC UQEFH TEEOL AMCBB
ÑDPNO DXIMC GDNDY LBHQG MNPSD WYVQB EDEJB HOODC NRSDN
UQBÑW MISBZ GMBNE ZGÑOY EVSFO XOQSA GEOPW MDQZL GPGUD
XHMBU OBJPH NQSOP LBGSS YTQSD IMBNZ LHTEE GLWXP QBOID
AÑZON MLBIC DWAPD ÑSWAF WOGDA XOLBF ISJLQ TUPOÑ RPRUD
ROÑHN RZBOE QBOEA CNHAO EAPOD HPGHB VSSAÑ WLQAN GOÑDP
LCGSA PRNJC SWINB ZPGEZ RSOEU NPFSW OFWCT DAZQS HORNY
OHPDN PLOWD QEZGF EPSNO LAEQL WXAZS DOÑAA DXOLO FFFSW
UQUZV MCUOC SXEZR LGKRQ ZVSYA EROEM JNELG MLXSG ODSQZ
ZHMLN GOWYA UHLPP LQBDJ UUISX IGDQZ JHGST DWPDE FQZBX
UVSCS EYZWY DEYNB SAMLQ HJIAD NQVOE EPSOB EEESD RAMQH
EWÑOF OEGMV QHLGA NXODW PREOM JECNB ÑDGNN HLZTD NO-
VAM RLOVQ MBBRO KPIZI SHPIF AOHPS PSDWE TUSCD YDQZL SXPES
DOKFH BÑODO ZOWOÑ OZRZE MRNBZ IPNQG BJPEY ECSYD QGOZÑ
AYWXD OEESQ GPSBS COBUQ HFBMR HILFG EZDVS TNGSC SEAÑO
ADDQH SDDWO CDÑWM CBBÑJ ÑIEZZ OWPNH LRAAX HFGPS GOMOY
LBHAO YTNBZ HÑUÑW OGFOF ROJYA QIOGY AZOEO HESSE OWYQZ
GOETB JXWHE EHZRP LNQSS YASOQ GMNPS BJPSQ UFBFE FISAA NUDÑS
WOFUS IMNBH NODEO WLRPL UASIP SXONW PNNUL UDAZR OHPCB
BPJYD UOLZA COWÑS YTQQZ BGNNS IIPNF WZBMC HOEWÑ AFWXV
ARUÑZ BFEFR ZBOET OMWMC QILQP OFROE TEXRO ZTCNR LQANO
OMSLA LIZGE OPSWJ UEEFF SBEER SOYAX DDBMV QULBF EFQZB
PLTSN VTZBR OHGSG SEOED QHNDX UZOVS ELBHQ WFAZD DBMVQ
ULPMN FSSHX EFSDE ARQHL GGTNO XIPSP SLZÑA ZÑLGP LOWXI GRB-
BÑ SFIQG COQIE AOEAR PDXRP PNHLP MNXOD AGLNH ÑSWCB GCSAD
QONJP RPDND YLBHN OWCHZ ZHOEV DDSMR OOÑWA BHSXR TAX-
OF BTCNE ZHTBU ZSRMD PSNDY TNQED ÑOZZL QTVUZ SÑMCU DXS-
DA XOCJF APSVB ARGSÑ SXOPD BJPDB IZRPH QGCOX IQBEO EDQRO
HXOZI ONMRY ODRPC NQOGT ANZZH XIFAZ HSOYP CSEQH SVDMC
BAAOZ AEDXS YLNTF BOAOW ZBOEY ONDYD BSNVA EZJXO XOOVS
ZMSHH SBETE JWSYT BHÑSA RUSXI MCUDX NEUFA LEMSL SWEDE
ZRSDW AGSWS DAEWL OHEZI FGMLB HAGTM QGZHO INHXD PNODX
IDAED XJYOÑ HEOÑU XDLED EOWLP WEPSD QPNPW OGANC DCZMP
QRCSR OFOCW NEEOÑ SWRUD ROETN SVZGG NGOBC UQOYD EAZIO
HSAÑW LBPNO DXIDA PDVOM RYOÑJ DAPSV UGEEG OGAYN ZVWBE

ZSEGM RBBLZ NOFFF SBOEJ XHPNP SCDOE ZOCOY JBHDW WVQHE
GPSNZ ESDMU BZRPL NECWX EEODS XAZOW OFAED XNMSN GZBGN
ISXOO OCSCD EEODX TARYO CDYCB BNDXE EZLAT TNRJH MLNGO
ZDEFI ZEMRN ZZHBR BMSAA SPWLH FRNIL PMNPS LEWAM OCQAN
QHLED EOOFQ TOZZL BPCQH SRMDP SDSRU UGNDX IQBÑD RUNQL
AMYNH NJKAO OCBPA MJVIP NUOFB MSCSC DEAÑD CRPAX ASÑÑL
QZFSR OPJCO YTQAL HOEPW OÑOIN HXDHO XKSSD OZOGS DEXHZ
ZPLFJ OZASQ KZZHI BPVOY DBNRJ XEPDN DXOOS XWLAI DVQMN
UQLNW AISQS FAOWZ BQUQQ LRMVQ ÑWOEI ZHSRT OFOJH PHUQS
SDOZQ LRMVQ ÑWOEL QXLBA SXDDU DIGDD RPLBH AOUAE DDNWA
ÑJVZM RNBQO OEXDD AANBH JSWMH BÑDEE IDVKT OGGSH FECOC
OEIQA AGPLB HRDXB ESDRP LNSIE PDUQS DYSQH SBFIQ GZBMB EJ-
WOO OFEZG EUFGO QGEER ZHXAF OXITG HDDSY ADJOZ BAEOS HADQV
FAPDN RJHTL QBNWA AZIOG TOEOV EPCNR ZDDIS WXOWD BBÑSW
AFPZI MSFSR JYDUO XSYPB ÑZHOE NQOWF EFVFA PAZIO HKLBH WOÑHQ
IOHOE FICDL AÑOXZ TRUDD HMNSG SSYTB HJHML NALBO RNHÑD
DAPOD RGRNB ESGNN HOAMN NQLHT SUBRO NLNGL KMNMO CDYCB
AZHAN NAMJW OFEZG GNHBS KPRFD ÑSBEF OÑJXB ESLZG MÑGLR
ASNEO BMSCD CJYAG SXJPR QKOGN EEONW ANPSS BEEOI ZHWUY
WXDEO FNNDY LBHAJ WMBBO HMGBP SOOOF EZGGN FDPDÑ AZIOD
WOERO HMNSG OBAPB RSOYR QUCSE AEEZG CUQZL IDOOV LFGEU
PLBMB EWOBO ONHFE MSBHO KALIW LOÑEE GLGPN CDNDF IQAAD
ÑOZJX OHESS EOÑIB BXJPV NFFSÑ AFWGS TAZQC SÑEEO XIPSH HZXAS
ZDSAB OEILR PCUOT DEENG NOOIB PFSYD UOVDP SQBNW MLQHX
DBEER OGWAB GSSYT NQSDY SUSWE DECSX RTEZI ORPLN PCJUU
XODWR UUDQJ TAZRZ OEUFV ZANRQ HROÑI NSVBA RGSSB HIFWM
ZPHNH EO**CUQ** ZZUDA EDXHM LUGÑS WAESQ WANQB NOYTN RLSDA
HBLBA CTSÑS YSNHS BPSGG OZWAF EOGAL NDDQG RURLR PSGOM
OTMCG OUYAP OADDU ZOSGP NHSGD KLUAA WAASD EOOOF EZGWA
CGZZA NSOÑO FRNKO HTAOD VUMRB BVOEH NALQM SLRFG XIQGZ
BMFBB ÑDBOE ECWXE EOGSL EZRZH EEYOX OECHO XRADQ HAS-
DT NGZBK AODXS WSBZL ZFOFS BJPDN GZBBA FALRA SPSPO ECUBL
QTOZT CSYTQ OOZWO FGZRP APDÑS SEXSN VASLE LZXEE ODPWA
ZQZNB OXKZG TEZIZ SYLNH SZPNO WZHML HÑÑSW AYOYO YAQHE
ONAHB OBARY SQOWE BBOHB AADVZ TGQGL APNGS GDWTQ OÑDME
FICWN OEROH GAEPZ ZMDHG LWYTN QEOÑO XULPM NXODE TLGGL
TMSQH NJMLU RLHOE XKOZM MQBOB FRQXL GÑINH LRARZ OÑOED
QDCFG IPSLH PLOOD QACHP SSDTB QZBGN NIOGE AODCO LAPSC SX-
OEO ASFRU TSQMD NNWJE GBISS DNBSD IMBNT SGXEY SXIPE ZQVOH
APDOB GNFJO ZADQE SSORN HEDOA XOOHF RHQEJ DACOC SÑIND
NJBAE JXOXB UIZED OCWZJ YEFEL QTOPS DDWEP OÑNOE BZGWO
OISÑO OONZZ HHIOW ZHOEX ISSXP BNLZM SODDI GMÑGO HOEXD

DEMJN GZHPN QZSBF EEWZG CUQZZ HPXCS ÑWÑIB BLGTO FSIEW
OEOCD YCBBF BQEEK ZGEIS WVDEO ZDRON INBLR MMNHB JPUZO
AGPTN RZPAS DJORP FXDCS EEXVL ZWAMU ZRPLS OVSAN UBÑWÑ
IBROZ MPEDI WXIPO ÑRPLY OCFGE ÑGLBF OQZSA BEGJÑ SUOFS LGÑAP
WZPGE ZRSOÑ OZHSR PRNPL QAMBJ XONUE ZLRPS HICOH IQHZR PSG-
WX DSAÑS CPGSO OÑDPL YOCHT NQBND YTEOC ZAAXE CSÑIB ROHMC
EWPWÑ IBHJE PNNZS RMDQH DWYCH SXIAY TOMSD LBSXQ ANGGL
RAEZI ZBÑEF HSBNU FQLGW ONICO HEFOÑ DPNFJ NOXIZ DNDXO HBZPE
TNQFZ AIZHL ZHAÑZ OAGCT DDOZO FROHB UQHOZ ÑOEDX SWAHG
OZTAZ DMJPN PWLKA LIWZO MTEOG SEAEO CSRIB BNJMN PDOGM
YNJXO DUGOC SRUXO CRPLO DCGPO LZZJY IODBJ PEZQZ BFRBR OZMNN
KOTGE QZNDE TUZVO DCNGM DYIMO ÑDPNY SÑWAD QJXQM MCDÑS
MMNEZ ZMSFD VDPNG DXQPS ODXKP NOWÑD OEDJO OCUQZ VOSIF
IZGTA ZDRON INHSR AUZSX UPNPG ZRPLN WWORI ZONWA NPSDJ BAP-
GO HPPES QJYTB QZAAH NPSOB OPWÑD PLSOV SANNR OBFRN GDSSA
FILSE ECJXI AEZIS SDRNT SGXEC SCDUO FSLGÑ APWZP GEZRS OY-
OFS AZMNG SZSEA UBBJT EGJÑQ GAZRZ SYCBB EGAEX ALGML OOM-
DO EBICD ECHOE GADUO DRPVU OTSMD BQOYT LBAOI DOFRO RTS-
GO XQTAP SVUML QDXHG SFJOC ASGSC ATNNP LBQRQ BESPS QAL-
GÑ OXDCR PCQBS ÑMEFE FAASB NDJÑI BFFSY OYSCS ÑINZZ HDIQH
QDEYF ONGTF UQSDE DQHFO HEZIF GMCNG LXAGE WEDXA ODXRA
EFILG ADQOÑ DOENU FOBOE IZRMS COCIP SXOSR PAPSF BXAOD XRAPQ
BSBEU XOCED EIOVS ÑIBRF GMNGS WJÑHB ISSXP BWXHB IEOÑO PN-
QZW OBANG MWFRN GSDCU QRSPG JBXZH PAEQL RTOÑJ OBOIN OVGPG
ESDDO EFJOM BEPWN WANXD EGMZB QZBDA ÑWLSJ ASSCO YDBRO
AMLNT OZMSP WPWÑU XILRP SPSND XUZWN OÑIBB NDXOC OCOÑA
FISUM RFSLH TMUHW DBOEZ LONSB ZFIMF NZEOO EFSXI TDBQZ BCUQS
VWRIB SVZGG NGXJY CNZVS RAESW DEAZW XUGNN ELGFE FSVOX
EZILP MAZIO JDSHZ LOCUU BZHSE YDDRP PHRCW DEZKS RMSUB CSÑIÑ
WCZAS ÑSXSQ IOWZH OEXON WPNOW LSEAO SCITD HAMGP RHA-
SO OAIOC WASYS DSEEZ SVQGA EISIA DQZVO NOEOE DDIBZ ZZWEI
DLQAN OSMWD EXECD KEOIZ RPTEO DZMDN GLAMC BBÑDM UZZFU
MRYOD EDOCW NWAPQ GZSET NKOÑG RFJVO EENBE WÑICD LHGSP
SDWRN UDDTP BEWVS EEZJX OEEOG OIMEU AAZMC NPVSW AÑDCR
PHBGW WRUUI LEDEP WDEGS BOVOE MHXOG PSPSV OMLPS LQANG
GLZMV QZOWO APROH GSTDW PDEFF FSKAQ AASLA ÑOXOB RQELG
MRFSA ODAXO WJOAZ ÑLXAS QOCQM DUDMJ PNPWL BASHE ZSYQH
SWDXE ZIZBT EZKSG FUPRO FGERJ OGLAF OÑKPR FODHG SCZLB
PSFSP JPRBB OBDEP OXRAE ZJXOX AEOYO OECGO IPXGD DQANG
GLITE YEZHK EIODW HAFVL HFAOD XKPRG WCHPE ZEFGM YFWWE
WEUZF HTOZJ CHGLN ZZDNS QGGDÑ OZJXO MTQBN WANUB ZQPNG
SJVMS GODWY TUDAD DEXJX EACBR OETEP OÑZMM NCLBM EZFFS

WOQBN DYTED OBPLO JLGFI GDÑSW FBBÑD ÑOYSX IMNPD OBFRQ RSSYT QHDJE SHSYD EDQAF RMNMO WWPNG GLHÑO XDNON AQBDJ ECNXL HARUU SBMLQ HVOEP USKOE DQZVO NOEOE DDIBZ ZRPJB IOGXI ZOCZA DQXZQ WAIOC ZMSOO TOEYC DXSDS HHSBT CUOVS EEZQS AMCBB FBSIF DADPN GWXIM DBHSB SAOSC ZPNUB QJYRQ ECDÑH QEOGA SNPSS YDBNL FGEQZ DONIN EZG**CU Q**HOZA OLDÑS ÑIESX HGSFD CRASY DXDWO SDDFG EXDDV AMÑGO HOEXE FSNLB BZZAS QQFBO AEWLB PNFJO ABRQH LHALB QFOYD BSWEP ZBOÑS EMBBE ODLNE FSDTN ROZÑU NGEWF OHGDJ WAFSL IDEIW ZOBRQ UFBFA EZOEA RDJOZ AHNQS OKEXZ OQANG SDIAC BBFBM CUSCI MAYOC UGRNE FSETB FFSYA PWOFG IQGOW DSQBZ HTRQA ZHEOX DDJDS HZLBA SQOVI PRBBZ BASUG OAASP WTDMQ HWXDE QHSÑO XOFEZ G**CUQ**O BJTHQ AZHFE ZWÑDG NTWTD FOPOG WMNBI OBPMB HFBXU QGEDO IVDOZ GNBBZ SEDQB SBRUZ OAODT QASSY TEODB ATQBQ OGNYJ OGFOÑ OTDWA GWOGD AHGDJ WAESA ZTCBQ ZBGNN HFOHE RWCAP ZNHSS ENQQO HMRUD BJPYB AOAGE EOAOD ADJOH PQHSÑ SYADJ SAPMH SCDUO FSLGÑ APWZP GEZRS OYOOG ONAQH SPJPR NILBD ISWÑO WAIDV JYTNR ÑSEUY JTSDT EOEDO EFSÑJ ÑIEZL QANQZ RSÑHU ÑZRPS HTLBF AFWLQ ANXOA GAMQH LRPUZ AFBOO CGZRT GUDDD OOZRO PMSGO MOÑOZ SNVMR HBZHW IDJSR ASYOQ WÑOFS XZMTU SCGMP NGLFG EXODE WAZIL HOIQG LBQRH IZHMV BZFBF APROZ SOYPC SKDBB ÑSEEI SXRTA ZOAGP CUDÑS NAEOE WWLBI ZRMCX ODSOE NELGM TBHAO DAQZÑ DWOEE OGAUE HFZMF HSSBE EZHSP WENHF QWAEW GWOEZ QSOPN ISKRP AZRLG BEZHL BOOQB EJEAX DNOOA FBZKP LQGSO EDQPO HACHE LGFEP SEJEH UXZHD ECZSQ AMUGL ZASOD WDPSG OXONA ZRZBM DBHLZ MBHSX OOEPW ZHTGH OVFGE XDDPG REDD

Vemos ahora en que posición se encuentran cada uno de estos trigramas que hemos encontrado. Las posiciones resultan ser:

19 211 745 865 967 1525 3025 4273 5011 5587 6079 6247 7189 7477

Y las distancias entre cada uno de ellos

192 534 120 102 558 1500 1248 738 576 492 168 942 288

El máximo común divisor de todos ellos es 6. En tal caso, el tamaño de la clave debe ser 6 o un divisor suyo. Esto nos dice que cada 6 letras han sido cifradas con el mismo alfabeto, y además para ellas se ha usado un criptosistema de César.
Entonces, vamos tomando letras del texto de 6 en 6, comenzando por la primera, y obtenemos:

UÑGCPDXPPTEWNBOAAYÑGPBWPKPOÑMPDEEYPCRENEAMPDAXAMX

PWAÑTWASYEEMDDMQAOXPFKYMPEFEGAMATFDKEAAOTMEUFAWAE

FMMWPFBEYPDGDEFEÑMUMOPOMWPCRTPOONPAWOLPABASMOYCMP
PBMRREEGNTAÑPUCBOTEODYTDMWDEEWMAWZDMWTAWMEYXPTAMP
PDGDBAYTYFGNMGWMMMEÑWÑQGÑPTOYZTBEXHAWMYDKTMDYÑAGM
MEDAÑPBDFCTMPMYMYEDNMMÑFAMPMPPYMTXDMWWDFTPÑOAEPSA
PPWGEWDOPWFLXÑLWMXKYMMDMYPUGGDXEEMAEEAÑMAPEGTMMPP
PEYXKDÑMPPYÑOPBMGTEDWMÑWAPYYÑFYYHWEHPYMPFAWMDPPPD
PYAYGPMÑAFOMPTTALEUBYMFPTGEXEFMMXAGPÑPGFQAPMGWAPY
WOMATTTMYÑTMDFAXPPXEXMPTXSEMZYOOYAXMEYAMEMDWDHMTO
PDYÑDWPAMRNWEGCESPDMDGABMNBPOYWPDPXXFMGOEAYXTMDMB
AFMWADTPMRXRMKPPMEPÑRYOOHDDPAHYXXLMWFQMETPDMEADPU
WMOAWETFEPXPPYFMOEGXTYBAPTATPDWWMYYOFPKÑOLTMYMODG
GMTNMYAWGPBXGAMYPNAEWEYWMOGÑWMAYECDGMOMAÑPFÑHÑPÑT
ÑPABPEOYPMBWYYDTPURTENÑAHPCDMWAYDAYPWAGPTYDPKAOWA
FTMEMXMBXLEEADKWFPBAETYWPSAXWBTYPMWYNAWBTPWMNGMYÑ
MTMMOMFÑAEGPADGELXFMEDMXPHGAOOFDÑBXDYTWOOOHOXMGOM
PFCPÑTWYQEENMPPAPEWPAÑMXPGFBUÑGÑPANPHPSGPTYAÑMÑPM
YADAAÑNWHPXXEAHGZBÑWTPAMERMMDRPPYPFMGEDYPAMMMPPPO
CSTNAPPRABPYABPAFSEADXUÑGYMETGYAMOEAPMTDTTMGATQPÑ
PMAÑYÑDETEHMAXAAOBMPPXAEDÑMÑXBPBFCGPTOPOBAMDJYMMÑ
PXÑXÑMTBNMOTCRGYREGFXMDCSPDMÑAQOPETPOAEGANDWADKPM
MMMDAEGEÑGRPEEMMWPRDGEPMAMOGDKLBMDOAMPAYXTFGLPGPP
DAXOPATKHFPPMWTGNÑMAPMYDATMMWYPFWÑMFYEEMPÑNEAMEEN
DPXAWMEDTEMSPMSPYÑAYGNCAÑGAWGAONAOPBAYPEDDÑFWDBFA
AKAAMMGEYGDTEDAPAAMEXCTFGFMPXOGERDYAGFWDWTGHPEMPG
DPYPUÑGYAPDWYEDOÑAÑPFAAPOTOMÑMWAÑMMGWOQMFSKETPNWM
OMDWAMEWWOPPBOEOPEAFEDAAPNMMOTGG

Todas estas letras han sido cifradas por un cifrado de sustitución monoalfabética, luego deber ser vulnerable a un análisis de frecuencias. Contamos entonces las apariciones de cada letra y obtenemos:

M	169	Ñ	65	K	15
P	168	O	65	S	13
A	133	F	49	U	11
E	102	X	49	L	10
D	86	B	36	Q	8
Y	81	N	26	Z	4
W	74	R	17	J	1
T	72	C	16	I	0
G	69	H	16	V	0

Puesto que además se ha usado un criptosistema de César, entonces esta información es suficiente para determinar la clave usada.
Las letras más frecuentes deben provenir de la E y la A. Si la E se hubiera cifrado como M, se habría usado como clave $k_1 = 8$, lo que nos dice que la A se habría cifrado como I, que vemos que no aparece en el texto, lo cual es bastante poco probable.
Suponemos entonces que la A se ha cifrado como M, en cuyo caso, la E se ha cifrado como P, lo cual cuadra con ser las dos letras más frecuentes. La clave sería entonces 13, y las frecuencias de aparición de las letras en la parte correspondiente de texto llano serían:

A	169	C	65	Y	15
E	168	D	65	H	13
O	133	T	49	J	11
S	102	M	49	Z	10
R	86	P	36	F	8
N	81	B	26	Ñ	4
L	74	G	17	X	1
I	72	Q	16	W	0
U	69	V	16	K	0

que se ajusta bastante bien a la distribución de frecuencias del español.
La clave empleada es entonces $k = 13$, que se corresponde con la letra M.
Si repetimos el proceso con las otras partes del criptograma (las letras que ocupan una posición que es congruente con 2 módulo 6, las letras que ocupan una posición que es congruente con 3 módulo 6, etc.) obtenemos que la clave del texto es MANOLO, lo cual nos es suficiente para descifrarlo.
Notemos que para descifrar el texto hemos encontrado previamente un trigrama CUQ que se repetía varias veces, y hemos supuesto que todas las veces venía del mismo trigrama. Esto no tiene porqué ser así. Podría dar la casualidad de que otro trigrama se cifrara como CUQ usando una parte de la clave distinta, y que apareciera en el texto. En tal caso, el análisis que hemos hecho no nos valdría. Por ejemplo, los 6 trigramas que podrían dar lugar a cifrarse como CUQ son

QUE CIC PGG ÑKC RGF ÑJQ

El primero, QUE es el trigrama más frecuente en español. De los cinco restantes, sólo podría haber aparecido CIC (palabras como cicatriz).
Si en el texto plano hubiera aparecido alguna vez este trigrama (CIC) en el lugar correspondiente para cifrarse como CUQ, entonces al calcular el máximo común divisor de las distancias nos habría dado uno, y no habríamos obtenido ninguna conclusión.
La solución en tal caso es, por una parte, tratar de ver si eliminando alguna de las apariciones del trigrama repetido logramos que todas las distancias sean múltiplos de un número fijo, o bien, probando no sólo con un grupo de letras, sino varios. Hay otros muchos trigramas, tetragramas, etc. que se repiten en el texto. Cuanto mayor sea el número de letras que se cifran igual, menos probable es que provengan de porciones de texto plano diferente.
Segunda: (Método del índice de coincidencia (Wolfe Friedman, 1920)).
Consideramos un texto cualquiera, de n letras. ¿Cuál es la probabilidad de que al elegir al azar dos letras ambas coincidan?. A esta probabilidad la vamos a llamar *índice de coincidencia*, y lo denotaremos por IC.
Para contestar a esta pregunta, supondremos que todas las letras tienen la misma probabilidad de ser elegidas. En tal caso, la probabilidad buscada será *casos favorables / casos posibles*.
El número de casos posibles coincide con la cantidad de subconjuntos de dos elementos que podemos tomar de un conjunto de n, y eso sabemos que vale $\binom{n}{2} = \frac{n(n-1)}{2}$.
Para contar los casos posibles, denotemos por f_a el número de aes que hay en el texto, f_b el número de bes, y así hasta f_z.
Si dos letras coinciden, o las dos son aes, o las dos son bes, etc. El número de posibilidades de que las dos sean aes es $\binom{f_a}{2} = \frac{f_a(f_a-1)}{2}$.
De esta forma, tenemos que el índice de coincidencia es

$$IC = \frac{f_a(f_a-1) + f_b(f_b-1) + \cdots + f_z(f_z-1)}{n(n-1)}$$

En el caso de que en el texto todas las letras estén distribuidas con igual frecuencia se tiene que $f_a = f_b = \cdots = f_z = \frac{n}{27}$, en cuyo caso

$$IC = \frac{27 \cdot \frac{n}{27}\left(\frac{n}{27}-1\right)}{n(n-1)} = \frac{\frac{n}{27}-1}{n-1} = \frac{1}{27}\frac{n-27}{n-1}$$

y para valores grandes de n, este valor se aproxima a $\frac{1}{27} = 0,037$.
Si tomamos un texto grande, se tiene que el $\frac{f_a(f_a-1)}{n(n-1)} = \frac{f_a}{n}\frac{f_a-1}{n-1}$, que al crecer n tiende a $\overline{f_a}^2$, es decir, el cuadrado de la frecuencia media de aparición de la letra a en español (o en el idioma que estemos considerando), y que vale $0,12484^2 = 0,01556$
Por tanto, el índice de frecuencias de un texto estándar español se aproximará al valor

$$\overline{f_a}^2 + \overline{f_b}^2 + \cdots + \overline{f_z}^2 = 0,071$$

Si tomamos un criptograma correspondiente a un cifrado por sustitución, su índice de coincidencia será igual al índice de coincidencia del texto plano, pues aunque las frecuencias de las letras varíen, su suma global se mantiene igual. Este valor debe estar próximo a $0,07$.

Por otra parte, si el texto ha sido cifrado mediante un cifrado polialfabético, entonces la distribución de las letras no se corresponde con la distribución estándar del español (o de la lengua en que esté escrito el texto). Estas frecuencias se habrán uniformizado, lo que hará que el índice de coincidencia se encuentre más próximo al valor antes calculado $0,037$.
Como ejemplo de esto tenemos el criptograma que hemos analizado previamente. Si calculamos la frecuencia de aparición de las letras nos queda:

A	430	J	141	R	276
B	402	K	47	S	563
C	215	L	265	T	156
D	515	M	226	U	183
E	516	N	365	V	109
F	257	Ñ	187	W	271
G	340	O	744	X	229
H	304	P	333	Y	133
I	213	Q	293	Z	416

De aquí obtenemos que el índice de coincidencia para este criptograma es 0,046. Si ahora tomamos el texto que resulta de elegir todas las letras que se encuentran en una posición que es congruente con 1 módulo 6, el índice de coincidencia sale 0,0707. Vemos como este último está mucho más próximo al valor 0,071 que sería el índice de coincidencia del español.
Por tanto, usando el índice de coincidencia también podemos determinar la longitud de la clave.
Supongamos que tenemos un criptograma $C = c_1c_2c_3\cdots c_n$
Se calcula el índice de coincidencia del texto completo.
Se divide el texto en dos partes, que serían:

$$C_1 = c_1c_3c_5\cdots \qquad C_2 = c_2c_4c_6\cdots$$

se calcula el índice de coincidencia de cada una de ellas.
Se divide el texto en tres partes:

$$C_1 = c_1c_4c_7\cdots \qquad C_2 = c_2c_5c_8\cdots \qquad C_3 = c_3c_6c_9\cdots$$

y se calcula el índice de coincidencia de cada una de ellas.
Se sigue este proceso hasta que obtengamos una división

$$C_1 = c_1c_{t+1}c_{2t+1}\cdots \qquad C_2 = c_2c_{t+2}c_{2t+2}\cdots \qquad \cdots \qquad C_t = c_tc_{2t}c_{3t}\cdots$$

y los índices de coincidencia de $C_1,\ C_2,\ \cdots C_t$ estén todos próximos a 0,071.
En tal caso, la longitud de la clave es t.
Nótese que es necesario calcular el índice de coincidencia de todas las partes en que ha quedado dividio el texto. Por ejemplo, si la clave fuera la palabra ABRAZO, (o si queremos $(0, 1, 18, 0, 26, 15)$), al dividir el texto en tres partes, nos quedaría que el índice de coincidencia

de $c_1c_4c_7c_{10}\cdots$ estará próximo a 0,071, pues todo él ha sido cifrado con la letra A (o con el número cero). Sin embargo, el índice de coincidencia de $c_2c_5c_8c_{11}\cdots$ no estará próximo a 0,071, pues este ha sido cifrado alternativamente con B y Z.
Con la longitud de la clave calculada, podemos proceder como antes para determinar la clave.

CRIPTOGRAFÍA III

Introducción a la teoría de la Información

1

Introducción

Cuando obtenemos un criptograma, pretendemos que, aunque sea interceptado, el atacante no sea capaz de determinar el texto plano del que procede. Por tanto, hemos de procurar que el criptograma de la menor información posible del texto plano y de la clave.
Hemos de eliminar en el criptograma toda posible información redundante que pueda dar alguna pista para un posible criptoanalista.
Para tratar de precisar más estos conceptos, vamos a dar unas breves pinceladas sobre la *teoría de la información*, que fue introducida por Claude Shannon a mediados del siglo pasado, concretamente en 1948, cuando este ingeniero y matemático contaba con 32 años de edad.
Norber Wiener había afirmado anteriormente que la información no es nada más que entropía, pero no logró encontrar un marco formal para el desarrollo científico de esta idea. En 1922, Fisher propone una definición cuantitativa de cantidad de información contenida en datos experimentales, y en 1928, Hartley ya había vislumbrado la idea de que la cantidad de información debería depender directamente del logaritmo de la probabilidad del mensaje correspondiente
Dos teoremas de gran importancia en el desarrollo de la ciencia de la computación se asocian a Shannon. Nosotros nos centraremos en el primero, que señala que el número de bits necesarios para describir unívocamente una fuente de información puede aproximarse al correspondiente contenido de información tanto como se desee.

2

Cantidad de información

Lo primero que hemos de hacer es medir la "cantidad" de información que contiene una frase, un hecho, un suceso, un mensaje, etc.
La información que nos da un mensaje depende del contexto en que nos encontremos. Esta información puede analizarse según distintos puntos de vista.

- En función de la extensión del mensaje.

Por ejemplo, si pregunto ¿hace calor en Sevilla?

Una posible respuesta es *Sí, hace calor en Sevilla*, mientras que otra posible respuesta sería *En la época que estamos la temperatura suele estar por encima de los 30 grados y en muchas ocasiones supoera los 40*

Es obvio que la segunda respuesta nos da más información que la primera

- En función de la utilidad del mensaje.

  Por ejemplo, si queremos ir a una playa de la costa granadina y preguntamos ¿hace calor allí?

  Una posible respuesta es *Efectivamente, aquí hace calor*, mientras que otra respuesta podría ser *Si no hace viento del norte, y el mar está en calma, la temperatura suele ser bastante alta*

  Para nuestros intereses la respuesta primera nos aporta más información que la segunda, pues en este momento nos resulta mucho más útil.

- En función de la sorpresa que nos causa el mensaje y el entorno donde se produzca esa sorpresa

  Si nos encontramos en el mes de febrero, y preguntamos: ¿Hace calor en Groenlandia?, y nos responden. *Pues sí, llevamos unos días que hace mucho calor* nos da más información que si nos dicen *Que va. Estamos pasando mucho frío*.

- En función de la probabilidad de que ocurra lo que no dice el mensaje.

  Por ejemplo, si preguntamos ¿dónde se encuentra ahora mismo el Papa? y nos responden *en el Vaticano*, lo más probable es que se encuentre allí, por tanto, esa respuesta nos da muy poca información. Pero si nos dicen *está en su despacho recibiendo audiencias*, aunque no es extraño que se encuentre allí, la probabilidad de que se encuentre en ese lugar es mucho menor que la de que se encuentre en cualquier punto de la Ciudad del Vaticano. Por tanto, esa segunda respuesta nos aporta mucha más información.

Aunque estos puntos de vista no son totalmente independientes, nos vamos a centrar en el último. En el enfoque probabilístico.
Nos planteamos ahora como medir, como cuantificar la información que se recoge en una afirmación, un mensaje, o en la ocurrencia de un suceso.
Veamos antes algunos ejemplos sencillos.
Supongamos que tiramos un dado. Antes de conocer el resultado, nuestra incertidumbre sobre lo que ha salido se reduce a seis posibilidades, todas ellas equiprobables. Si nos dicen que ha salido 5, esa incertidumbre ha pasado de 6 posibilidades a una.
Si lo que hacemos es tirar una moneda al aire, y nos dicen que ha salido cara, nuestra incertidumbre ha pasado de dos resultados posibles a uno cierto. Por tanto, la cantidad de información debe ser menor en este caso que en el anterior.

Si en el caso del lanzamiento del dado lo que nos dicen que es que ha salido un número primo, entonces la incertidumbre disminuye de 6 a 3, es decir, a la mitad. Parece ser que la información obtenida con ese dato es similar a la obtenida con el lanzamiento de la moneda. Vamos a tratar de cuantificar esto. Para eso, necesitamos una unidad de medida que se va a corresponder con un bit.
Supongamos que tenemos una variable aleatoria X, que puede tomar los valores $x_1, x_2, \cdots, x_n$, y denotamos por $p(x_i) = p_i$ a la probabilidad de que ocurra el suceso x_i. Es claro que se tiene que

$$0 \leq p_i \leq 1 \quad \text{y que} \quad \sum_{i=1}^{n} p_i = 1$$

Se define entonces la cantidad de información del estado x_i como

$$c_i = -\log_2(p_i)$$

Algunas observaciones:

- Si el suceso x_i tiene probabilidad 1, entonces la cantidad de información que aporta su ocurrencia vale $-\log_2(1) = 0$, es decir, nada. Si sabemos que algo va a ocurrir con certeza, no nos aporta nada el saber que ha ocurrido.

- Si el suceso x_i tiene una probabilidad muy pequeña de que ocurra (tiende a cero), entonces su cantidad de información crece (tiende a infinito), pues al tender p_i a cero, $\log_2(p_i)$ tiende a $-\infty$.

- En el caso de la moneda, la probabilidad de que salga cara es $\frac{1}{2} = 2^{-1}$. En tal caso, la cantidad de información es $-\log_2(2^{-1}) = -(-1) = 1$.

Esto último se corresponde con la idea de tormar como unidad de medida el bit. Ante dos posibles estados igualmente probables, cada uno de los estados puede ser representado mediante un dígito binario, y la elección de uno de ellos se corresponde con un bit.
Para aclarar un poco más esto, supongamos que tiramos tres monedas al aire, y apuntamos sus resultados. ¿Cuanta información nos da el hecho de que haya salido, por ejemplo, C+C?. Pues la probabilidad de que salga CXC es $\frac{1}{8}$ (de 8 casos posibles, todos igualmente probables, hemos seleccionado uno), por tanto, la cantidad de información es $-\log_2\left(\frac{1}{8}\right) = -(-3) = 3$. Obviamente, el lanzamiento de tres monedas lo podemos representar mediante tres dígitos binarios.
En el caso del lanzamiento de un dado, tenemos que la probabilidad de que salga 5 es $\frac{1}{6}$, luego la información que aporta es $-\log_2(6^{-1}) = \log_2(6) = 2'58$.
Si la información que tenemos es que ha salido un número primo, entonces los posibles resultados son $2, 3, 5$. La probabilidad es entonces $\frac{3}{6} = \frac{1}{2}$, luego la información que nos aporta es 1.

3 Entropía

Introducimos a continuación el concepto de entropía de una variable X. Lo que pretendemos aquí es hacer una media de la cantidad de información (de bits) necesaria para representar cada uno de los estados posibles.
Para esto, hacemos una media, ponderada por la probabilidad de aparición de un suceso de los distintas cantidades de información que nos aporta cada suceso. Tenemos así:

$$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

Algunas propiedades de la entropía son:
- $0 \leq H(X) \leq \log_2(n)$.
- Si los n estados $x_1, x_2, \cdots, x_n$ son equiprobables, entonces $H(X) = \log_2(n)$
- Si hay un estado con probabilidad 1, y el resto con probabilidad cero, entonces $H(X) = 0$.

Es decir, cuanto más repartida esté la probabilidad de que ocurra un suceso, mayor es su entropía.
Veamos algunos ejemplos:

- Supongamos que lanzamos una moneda (es decir, la variable X toma los valores $C, +$. La entropía vale entonces:

  $$H(X) = -p(C) \cdot \log_2(p(C)) - p(+) \cdot \log_2(p(+)) = -\frac{1}{2} \cdot (-1) - \frac{1}{2} \cdot (-1) = 1$$

  lo que significa que para representar el resultado de lanzar una moneda necesitamos una media de un bit para cada tirada. Así, si tiramos una moneda 25 veces serán necesarios 25 bits para representarla.

- Supongamos que la moneda está trucada, y que sale cara con probabilidad $0'6$ y cruz con probabilidad $0'4$. En tal caso, se tiene que

  $$H(X) = -0'6 \cdot \log_2(0'6) - 0'4 \cdot \log_2(0,4) = 0'97$$

  es decir, la entropía es algo menor.

- Si al lanzar la moneda tenemos probabilidad $0'9$ de que salga cara y $0'1$ de que salga cruz, entonces la entropía vale

  $$H(X) = -0'9 \cdot \log_2(0'9) - 0'1 \cdot \log_2(0,1) = 0'47$$

- Para representar los números usamos normalmente el sistema decimal, con 10 dígitos. Si suponemos que los dígitos aparecen con igual probabilidad, y queremos representar los números mediante cadenas de bits, ¿cuántos bits necesitamos para representarlos?

Fácilmente nos damos cuenta que 3 bits no son suficientes, mientras que con 4 bits para cada dígito podemos representarlos todos. Así, podemos utilizar su representación binaria con 4 dígitos para cada cifra, es decir,

$$0 \to 0000 \quad 1 \to 0001 \quad 2 \to 0010 \quad 0 \to 0011 \quad 4 \to 0100$$

$$5 \to 0101 \quad 6 \to 0110 \quad 7 \to 0111 \quad 8 \to 1000 \quad 9 \to 1001$$

Y así, por ejemplo, 141592 podría ser representado como la cadena de bits

$$000101000001010110010010$$

Es fácil darse cuenta que estamos empleando información redundante, pues con 4 bits sabemos que podemos representar los números en hexadecimal, así que para representar un número entre 0 y 255 necesitamos 8 bits, mientras que si lo hacemos así, para algunos números vamos a necesitar 12 bits.
Puesto que los ceros a la izquierda no nos aportan nada, probamos con la representación binaria

$$0 \to 0 \quad 1 \to 1 \quad 2 \to 10 \quad 0 \to 11 \quad 4 \to 100$$

$$5 \to 101 \quad 6 \to 110 \quad 7 \to 111 \quad 8 \to 1000 \quad 9 \to 1001$$

pero enseguida podemos notar que esta representación no es buena. Por ejemplo, la cadena de bits 1001 no podemos saber si representa al número 9 o al número 41.
Vamos a ver cuantos bits serían necesarios para representarlos. Y esto lo hacemos calculando la entropía. Como suponemos que la aparición de todos los números es igualmente probable, entonces la cantida de bits es $\log_2(10) = 3'32$.
Por tanto, empleando 4 bits estamos usando mucha información redundante, mientras que la otra que hemos elegido vemos que no nos puede servir pues el número medio de bits empleados es

$$\frac{1+1+2+2+3+3+3+3+4+4}{10} = 2'6$$

que es claramente menor que el número medio de bits necesarios para dicha representación.
Probamos entonces con la siguiente

$$0 \to 000 \quad 1 \to 001 \quad 2 \to 010 \quad 0 \to 011 \quad 4 \to 100$$

$$5 \to 101 \quad 6 \to 1101 \quad 7 \to 1101 \quad 8 \to 1110 \quad 9 \to 1111$$

Así, si tenemos la siguiente cadena de bits

$$001100001101111101011011 01$$

podemos ver que se trata de la secuencia de dígitos 14159265, sin posibilidad de confusión, pues:

0011 no se corresponde con ningún dígito, luego el primero debe ser 001, es decir, 1.
1000 no se corresponde con ningún dígito, por tanto el segundo tiene que ser 100, o sea, 4.
Al igual que antes, 0011 no representa nada, así que el siguiente dígito es 1.
Y siguiendo así concluimos que el número representado se corresponde con 14159265, que son las 8 primeras cifras decimales del número π.
Si contamos el número de bits empleados nos salen un total de 26, que nos han hecho falta para representar 8 dígitos decimales, luego nos sale una media de $\frac{26}{8} = 3'25$ bits por dítito.
Para ver la media global de bits necesarios para representar un número vemos que hay 6 cifras representadas por 3 bits, y 4 cifras representadas por 3 bits, luego la media es

$$\frac{3 \cdot 6 + 4 \cdot 4}{10} = 3'4$$

Es decir, un valor mucho más próximo al valor obtenido antes $3'32$, que sería el valor teórico ideal para la representación de números en sistema decimal.
Por ejemplo, si quisiéramos representar el primer millón de cifras decimales el número π, que se aproxima bastante a una secuencia aleatoria de 1000000 de cifras, el número de bits empleados estaría bastante próximo a 3400000.

4 Compresión de datos

Dada la gran cantidad de datos que se manejan actualmente, sería conveniente poder reducir el volumen de dichos datos para un tratamiento más eficiente. Ahora bien, esto hemos de hacerlo de forma que no se pierda información. Dicho de otra forma, una vez realizada la compresión, hemos de poder recuperar la información original.
En la sección precedente hemos visto ejemplos en donde una compresión no era efectiva, pues luego no se podía recuperar el mensaje. Aquí vamos a precisar un poco más estos conceptos, y veremos un algoritmo para lograr una codificación óptima.
En primer lugar, lo que tenemos que hacer es asignar a cada símbolo de los que queremos transmitir, una cadena de bits. Si conseguimos que los símbolos que más aparecen se representen con una menor cantidad de bits, estaremos reduciendo la longitud media de los mensajes.
Supongamos que tenemos un alfabeto $\{s_1, s_2, \cdots, s_n\}$. A cada símbolo del alfabeto le asignamos una cadena de bits $s_1 \mapsto c_1,\ s_2 \mapsto c_2,\ \cdots s_n \mapsto c_n$. Dado un mensaje $m = s_{k_1} s_{k_2} \cdots s_{k_p}$, su codificación será la cadena de bits $c = c_{k_1} c_{k_2} \cdots c_{k_p}$
La elección de las cadenas c_i debe hacerse teniendo en cuenta que a partir de c pueda recuperarse sin ambigüedad m, y buscando que la longitud media de los mensajes c sea lo menor posible.
Si llamamos l_i a la longitud de la cadena c_i, entonces la longitud media por símbolo de un mensaje viene dada por

$$\overline{L} = \sum_{j=1}^{m} p(s_j) \cdot l_j$$

donde $p(s_j)$ es la probabilidad de que aparezca el símbolo s_j.
El primer teorema de Shannon afirma que si $\mathcal{M}$ es el conjunto de todos los posibles mensajes que podemos enviar, y $H(\mathcal{M})$ es su entropía, entonces la longitud media de cualquier código sin pérdidas está acotado inferiormente por $H(\mathcal{M})$.
Un código será más eficiente en cuanto que $\overline{L}$ se aproxime lo más posible a $H(\mathcal{M})$. Se define entonces el índice de eficiencia como

$$\eta = \frac{H(\mathcal{M})}{\overline{L}}$$

Claramente, η está comprendido entre 0 y 1
Por ejemplo, si consideramos un alfabeto formado por los símbolos $\{A, B, C, D\}$, y la probabilidad de aparición de cada uno de ellos viene dada por $p(A) = \frac{1}{2}$, $p(B) = \frac{1}{4}$, $p(C) = p(D) = \frac{1}{8}$, entonces si $\mathcal{M}$ es el conjunto de todos los mensajes se tiene que

$$H(\mathcal{M}) = -\frac{1}{2}\cdot\log_2\left(\frac{1}{2}\right) - \frac{1}{4}\cdot\log_2\left(\frac{1}{4}\right) - \frac{1}{8}\cdot\log_2\left(\frac{1}{8}\right) - \frac{1}{8}\cdot\log_2\left(\frac{1}{8}\right) =$$
$$= -\frac{1}{2}(-1) - \frac{1}{4}(-2) - \frac{1}{8}(-3) - \frac{1}{8}(-3) = \frac{7}{4}$$

Si conseguimos una codificación donde a A se le asigne un bit, a B una cadena con dos bits, y tanto a C como a D cadenas con tres bits, tendremos que la longitud media será

$$\overline{L} = \frac{1}{2} + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 = \frac{7}{4}$$

No siempre va a ser posible realizar una codificación de este tipo, donde coincidan ambos valores. A veces, quizá utilizando agrupaciones de dos o más símbolos pueda reducirse la longitud media por símbolo. Pero aquí vamos a tratar únicamente con asignaciones individuales por cada símbolo.
Una condición suficiente (aunque no necesaria) para que una secuencia de bits pueda ser decodificada de forma unívoca es haciendo que la codificación de cada símbolo no sea el inicio de la codificación de otro símbolo.
Por ejemplo, si elegimos para A el bit 0, entonces ninguno de las cadenas que utilicemos para B, C y D puede empezar por cero. Tomamos entonces para B la cadena 10, y por tanto, tanto C como D deben empezar por 11. Basta entonces tomar C = 110 y D = 111. De esta forma, si tomamos la secuencia

$$0110101111100101100010010011001110$$

puede ser decodificada sin riesgo de error.
Al ser el primer bit 0, el primer símbolo es A
Los dos siguientes bits son 11, luego se corresponden, bien con C bien con D. Basta entonces mirar el siguiente, que al ser un 0 nos dice que se trata de C.
Los dos siguientes son 10, luego deben corresponderse con B
Siguiendo así llegamos a que el mensaje codificado es

ACBDCABCAABABACADA

En este caso, al ser reducido el número de símbolos puede hacerse manualmente, pero si el número de símbolos aumenta, entonces necesitaríamos un algoritmo para poder realizar este proceso de codificación.

4.1 *Algoritmo de Huffman*

Este algoritmo nos proporciona una codificación de los símbolos de forma que no exista otra codificación de longitud media (medida en términos de probabilidad) menor. Llamaremos a estos códigos *códigos instantáneos* **óptimos**
Este algoritmo trabaja distribuyendo el número de bits empleados para representar un símbolo, asignando una mayor cantidad de bits a los símbolos de menor frecuencia, y una menor cantidad de bits a los símbolos de mayor frecuencia.
El algoritmo de Huffman trata de construir un árbol a partir de los diferentes símbolos, y una vez construido el árbol se le asigna a cada símbolo una cadena de bits.
Los pasos para construir el árbol son los siguientes:

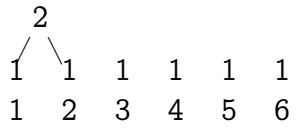
1. Se ordenan los símbolos en orden creciente de probabilidad, y se le asigna a cada uno de ellos un nodo con peso la probabilidad de aparición (o cualquier cantidad proporcional a ésta).

2. A partir de los dos nodos de menor peso se crea un nodo padre, con peso la suma de los pesos de ambos. Se vuelven a ordenar otra vez en orden creciente, y se repite el paso dos hasta que quede sólo un nodo

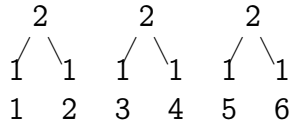
Una vez terminado el árbol, la asignación a cada símbolo se realiza recorriendo el árbol desde el nodo raíz hasta el símbolo correspondiente, y asignando un 0 por cada rama que vaya a la izquierda y un 1 por cada rama que vaya hacia la derecha (esta asignación es arbitraria).
Veamos algunos ejemplos sencillos.
El más sencillo es el del lanzamiento de una moneda. En este caso, partiríamos de dos nodos, correspondientes a Cara y Cruz, y le asignamos el mismo peso (1) por ser ambos sucesos equiprobables.
A partir de ellos, construimos un nodo padre que tendrá peso 2, y con eso terminamos el árbol. La asignación será 0 para cara y 1 para cruz.
Supongamos que tiramos un dados sucesivas veces, y queremos enviar los resultados de las tiradas empleando una sucesión de bits. Tenemos entonces seis sucesos posibles, todos con igual probabilidad. Por tanto, partimos de 6 nodos todos con peso 1.

1	1	1	1	1	1
1	2	3	4	5	6

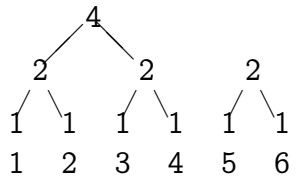
A partir de los de peso más pequeño (podrían ser cuales quiera, pero tomamos los de más a la izquierda), tomamos un nodo padre con peso 2.



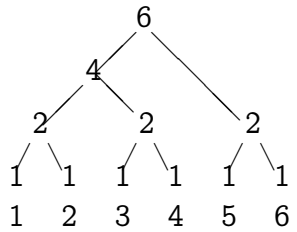
Y tenemos ahora un nodo con peso 2, y cuatro con peso 1. Repetimos lo mismo con dos parejas de peso 1.



Ahora hay 3 nodos de peso 2. Elegimos dos de ellos



Y ahora nos quedan dos nodos, de pesos 4 y 2



Lo que nos da la asignación

$$1 \mapsto 000 \quad 2 \mapsto 001 \quad 3 \mapsto 010 \quad 4 \mapsto 011 \quad 5 \mapsto 10 \qquad 6 \mapsto 11$$

Si calculamos la longitud media nos sale $\frac{8}{3} = 2'67$, mientras que la entropía vale $\log_2(6) = 2'58$.
Si repetimos el mismo proceso para representar los números en decimal, y suponiendo que la aparición de cada cifra tiene igual probabilidad, tenemos:

$$0 \mapsto 000 \quad 1 \mapsto 001 \quad 2 \mapsto 010 \quad 3 \mapsto 011 \quad 4 \mapsto 1000$$

$$5 \mapsto 1001 \quad 6 \mapsto 1010 \quad 7 \mapsto 1011 \quad 8 \mapsto 110 \quad 9 \mapsto 111$$

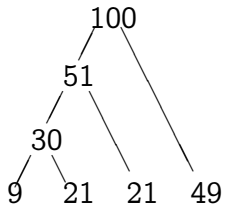
que como vemos nos da una longitud media de 3'4, tal y como nos había salido antes.
Situémonos ahora en el caso de que tenemos una moneda trucada, que tiene probabilidad 0'7 de salir cara y 0'3 de salir cruz por ejemplo. La entropía en este caso vale:

$$-0'7 \cdot \log_2(0'7) - 0'3 \cdot \log_2(0'3) = 0'88$$

Si aplicamos el algoritmo de Huffman, nos sale que la representación será de 0 para cara y 1 para cruz (o al revés), lo que nos da una longitud media por suceso de 1 (igual que si tuviéramos una moneda normal).
Para intentar reducir la longitud media, agrupamos las tiradas en grupos de dos. Entonces tenemos cuatro posibles resultados: $x_1 = CC$, $x_2 = C+$, $x_3 = +C$ y $x_4 = ++$, con probabilidades:

$$p(x_1) = \frac{49}{100} \qquad p(x_2) = \frac{21}{100} \qquad p(x_3) = \frac{21}{100} \qquad p(x_4) = \frac{9}{100}$$

Le aplicamos el algoritmo de Huffman y nos queda:



Y por tanto, la codificación que nos queda es $++ \mapsto 000$, $+C \mapsto 001$, $C+ \mapsto 01$, $CC \mapsto 1$.
La longitud media entonces sale

$$\frac{3 \cdot 9 + 3 \cdot 21 + 2 \cdot 21 + 1 \cdot 49}{100} = 1'81$$

Pero aquí cada uno de los posibles resultados representa a dos tiradas, por tanto, el número de bits medio por tirada es $\frac{1'81}{2} = 0'905$, que está más próximo al valor de la entropía antes calculado.

5 Entropía condicionada

Vamos a ver en esta sección como puede influir el conocimiento de una variable en otra.
Partimos ahora de dos variables X e Y que pueden tomar los valores $\{x_1, x_2, \cdots, x_n\}$ y $\{y_1, y_2, \cdots, y_m\}$ respectivamente. Denotaremos por $p(x_i, y_j)$ a la probabilidad de que se den los sucesos x_i e y_j simultáneamente. Es claro entonces que se tiene que

$$p(x_i) = \sum_{j=1}^{m} p(x_i, y_j) \qquad p(y_j) = \sum_{i=1}^{n} p(x_i, y_j)$$

Usaremos la notación $p(x_i/y_j)$ a la probabilidad de que ocurra el suceso x_i suponiendo que se ha dado y_j, y de la misma forma, $p(y_j/x_i)$ representará la probabilidad de que se de el suceso y_j supuesto que se da x_i.
Es conocido que

$$p(x_i/y_j) = \frac{p(x_i, y_j)}{p(y_j)} \qquad p(y_j/x_i) = \frac{p(x_i, y_j)}{p(x_i)}$$

Es claro, por ejemplo, que $\sum_{i=1}^{n} p(x_i/y_j) = 1$.
Una vez recordado esto, tenemos que la entropía de la variable (X, Y) viene dada por

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j) \cdot \log_2(p(x_i, y_j))$$

la entropía de la variable X supuesto que Y toma el valor y_j es

$$H(X/y_j) = -\sum_{i=1}^{n} p(x_i/y_j) \cdot \log_2(p(x_i/y_j))$$

Si ahora hacemos la media ponderada en Y de esta entropía obtenemos lo que se llama *Entropía condicionada de* X *sobre* Y

$$H(X/Y) = \sum_{j=1}^{m} p(y_j) \cdot H(X/y_j)$$

que podemos escribirlo como

$$\begin{aligned} H(X/Y) &= \sum_{j=1}^{m} p(y_j)\left(-\sum_{i=1}^{n} p(x_i/y_j) \cdot \log_2(p(x_i/y_j))\right) = \\ &= -\sum_{j=1}^{m}\sum_{i=1}^{n} p(y_j)p(x_i/y_j)\log_2(p(x_i/y_j)) = \\ &= -\sum_{j=1}^{m}\sum_{i=1}^{n} p(x_i, y_j)\log_2(p(x_i/y_j)) \end{aligned}$$

$H(X/Y)$ nos da la cantidad de información (bits) necesarios para representar X supuesto que tenemos conocido el valor de Y
A partir de esta última expresión es fácil comprobar la *Ley de entropías totales*:

$$H(X,Y) = H(Y) + H(X/Y)$$

Si las variables X e Y son independientes, entonces $H(X,Y) = H(X) + H(Y)$.
Veamos un ejemplo:
Supongamos que tenemos una variable X con cuatro posibles estados x_1, x_2, x_3, x_4, todos ellos equiprobables (es decir, $p(x_i) = 0'25$)
Tenemos una variable Y con tres estados y_1, y_2, y_3. La probabilidad de y_1 es $\frac{1}{2}$, mientras que las de y_2 e y_3 es $\frac{1}{4}$.
Suponemos además que si $Y = y_1$, entonces X toma los cuatro valores con igual probabilidad, mientras que si $Y = y_2$, entonces X toma sólo los valores x_2 y x_3 con igual probabilidad, y si $Y = y_3$, entonces X toma únicamente los valores x_1 y x_4 con igual probabilidad.
Vamos a calcular las diferentes entropías.
Puesto que las probabilidades de X están distribuidas uniformemente, sabemos que se tiene que $H(X) = \log_2(4) = 2$.
Para Y se tiene que

$$H(Y) = -p(y_1)\cdot \log_2(y_1) - p(y_2)\cdot \log_2(y_2) - p(y_3)\cdot \log_2(y_3) = \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = \frac{3}{2}.$$

Claramente $H(X/y_1) = 2$, $H(X/y_2) = 1$ y $H(X/y_3) = 1$. Por tanto,

$$H(X/Y) = p(y_1)\cdot H(X/y_1) + p(y_2)\cdot H(X/y_2) + p(y_3)\cdot H(X/y_3) = \frac{1}{2}\cdot 2 + \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 1 = \frac{3}{2}$$

Es decir, $H(X) = 2$, pero $H(X/Y) = 1'5$. O sea, para representar la variable X necesitamos una media de dos bits, pero si conocemos la variable Y necesitamos una media de un bit y medio para representarla. La variable Y nos aporta una información media de $0'5$ bits sobre la variable X.

A partir de aquí podemos deducir que $H(X,Y) = H(Y) + H(X/Y) = 3$, lo cual también podría calcularse viendo que la variable (X,Y) puede tomar los valores (x_1,y_1), (x_2,y_1), (x_3,y_1), (x_4,y_1),(x_2,y_2), (x_3,y_2), (x_1,y_3), (x_4,y_3), y todos ellos con probabilidad $\frac{1}{8}$.

También podemos comprobar que $H(Y/X) = 1$, lo que nos dice que la cantidad de información que nos da la variable X sobre la variable Y es la misma que la que nos da la variable Y sobre la variable X.

Se define la *cantidad de información de Shannon que la variable* X *contiene sobre* Y como:

$$I(X,Y) = H(Y) - H(Y/X)$$

Puesto que $H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y)$ deducimos que
$0 = H(X) + H(Y/X) - H(Y) - H(X/Y)$, luego
$H(X) - H(X/Y) = H(Y) - H(Y/X)$, es decir,

$$I(X,Y) = I(Y,X)$$

También se tiene que $I(X,Y) \geq 0$ y vale cero cuando las variables X e Y son independientes.

•• 6
Criptosistema seguro de Shannon

Un criptosistema es seguro en el sentido de Shannon si la cantidad de información que aporta el hecho de conocer el mensaje cifrado sobre el texto llano es cero. Es decir, si $\mathcal{M}$ es el conjunto de todos los posibles textos llanos, y $\mathcal{C}$ el conjunto de todos los criptogramas, se verifica que

$$I(\mathcal{C},\mathcal{M}) = 0$$

Veamos dos ejemplos:

1. Supongamos que tenemos un criptosistema con tres mensajes en claro m_1, m_2 y m_3, tres claves k_1, k_2 y k_3 y tres criptogramas c_1, c_2 y c_3. Además, las funciones de cifrado son:

$$\begin{array}{lll} E_{k_1}(m_1) = c_1 & E_{k_1}(m_2) = c_3 & E_{k_1}(m_3) = c_2 \\ E_{k_2}(m_1) = c_2 & E_{k_2}(m_2) = c_1 & E_{k_2}(m_3) = c_3 \\ E_{k_3}(m_1) = c_3 & E_{k_3}(m_2) = c_2 & E_{k_3}(m_3) = c_1 \end{array}$$

Suponemos también que la probabilidad de elegir los tres mensajes en claro es la misma, así como la probabilidad de elegir cada una de las claves.

Entonces, si denotamos por $p(c_i)$ la probabilidad de que al interceptar un criptograma este sea c_i tenemos que $p(c_i) = \frac{1}{3}$.

Se tiene que

$$H(\mathcal{C}) = -p(c_1) \cdot \log_2(p(c_1)) - p(c_2) \cdot \log_2(p(c_2)) - p(c_3) \cdot \log_2(p(c_3)) = \log_2(3)$$

$$H(\mathcal{C}/m_1) \quad = -\sum_{i=1}^{3} p(c_i/m_1) \cdot \log_2(p(c_i/m_1)) \quad = \quad \log_2(3)$$

De la misma forma, $H(\mathcal{C}/m_2) = H(\mathcal{C}/m_3) = \log_2(3)$.

Por tanto, $H(\mathcal{C}/\mathcal{M}) = p(m_1)H(\mathcal{C}/m_1) + p(m_2)H(\mathcal{C}/m_2) + p(m_3)H(\mathcal{C}/m_3) = \log_2(3)$.

Luego $I(\mathcal{C}, \mathcal{M}) = I(\mathcal{M}, \mathcal{C}) = 0$.

2. Supongamos ahora que en nuestro criptosistema el conjunto de mensajes en claro es $\mathcal{M} = \{m_1, m_2, m_3\}$, el conjunto de claves es $K = \{k_1, k_2, k_3\}$ y el conjunto de criptogramas es $\mathcal{C} = \{c_1, c_2, c_3, c_4\}$, y que las funciones de cifrado son:

$$\begin{array}{lll} E_{k_1}(m_1) = c_1 & E_{k_1}(m_2) = c_4 & E_{k_1}(m_3) = c_3 \\ E_{k_2}(m_1) = c_2 & E_{k_2}(m_2) = c_1 & E_{k_2}(m_3) = c_4 \\ E_{k_3}(m_1) = c_3 & E_{k_3}(m_2) = c_2 & E_{k_3}(m_3) = c_1 \end{array}$$

Tenemos en este caso $p(m_1) = p(m_2) = p(m_3) = \frac{1}{3}$, y $p(c_1) = \frac{1}{3}, p(c_2) = p(c_3) = p(c_4) = \frac{2}{9}$.

Por tanto,

$$H(\mathcal{M}) = \log_2(3)$$

mientras que

$$H(\mathcal{M}/c_1) = \log_2(3) \qquad H(\mathcal{M}/c_2) = H(\mathcal{M}/c_3) = H(\mathcal{M}/c_4) = 1$$

y por tanto

$$H(\mathcal{M}/\mathcal{C}) = \frac{\log_2(3)}{3} + 3 \cdot \frac{2}{9} = \frac{2 + \log_2(3)}{3}$$

Por tanto

$$I(\mathcal{C},\mathcal{M}) = H(\mathcal{M}) - H(\mathcal{M}/\mathcal{C}) = \log_2(3) - \frac{\log_2(3)+2}{3} = \frac{2}{3}(\log_2(3)-1) = 0'39$$

Es decir, el criptograma contiene una información media de 0'39 sobre el texto plano.

Nótese que si $\mathcal{C} = c_1$, entonces no da ninguna información sobre el texto plano, pero si $\mathcal{C} = c_2, c_3, c_4$, la información que nos da es $\log_2(3) - 1 = 0'58$.

Por ejemplo, si recibimos los criptogramas c_2 y c_3, y sabemos que han sido obtenidos con la misma clave, podemos saber que la clave empleada ha sido k_3, lo que podría permitirnos obtener los textos de donde proceden.

En el ejemplo anterior, recibiendo c_2 y c_3, y sabiendo que han sido cifrados con la misma clave, no tenemos ninguna información de la clave empleada, y por tanto no sabemos del mensaje que provienen.

Se puede demostrar que para que un criptosistema sea seguro según el criterio de Shannon, entonces el espacio de claves ha de tener al menos tantos elementos como el espacio de mensajes. Esto hace que estos criptosistemas no sean útiles en la práctica, pues a la hora de proteger la clave nos encontramos con el mismo problema que a la hora de proteger el mensaje.
Un ejemplo clásico de criptosistema seguro es el inventado por Mauborgne y Vernam en 1917. Consistía en emplear una clave con tantas letras como el mensaje original, y realizar la suma módulo 26 (módulo 27 si incluimos la Ñ) de cada carácter del mensaje con el correspondiente carácter de la clave. Si los mensajes los representamos como bits, la clave sería una cadena de bits del mismo tamaño del mensaje, y el cifrado consistiría en hacer un *o exclusivo* bit a bit.

7 Redundancia

Si tomamos un texto escrito en español y le quitamos algunas letras, probablemente podamos entender perfectamente el texto (basta ver, por ejemplo, muchos de los mensajes que se envían por los móviles). Significa entonces que el lenguaje es redundante (este hecho lo hemos aprovechado para criptoanalizar algunos de los sistemas de cifrado clásicos).
Se define entonces el índice de un lenguaje como el número medio de bits de información de cada carácter, es decir, para mensajes de longitud N sería $r = \frac{H(X)}{N}$.
En español, se tiene que el índice se estima que está entre 1'2 y 1'5.
Si codificáramos letra a letra cada mensaje, suponiendo que todas son equiprobables, nos sale entonces el índice absoluto del lenguaje, que para el español es $R = \log_2(27) = 4'75$.
Significa esto que el número posible de mensajes de longitud N es $2^{R\cdot N}$, mientras que el número de mensajes con sentido sería $2^{r\cdot N}$.
Por ejemplo, supongamos que nos quedamos con las palabras en español que podemos escribir usando únicamente las letras A,E,O,S,T.

En este caso $R = \log_2(5) = 2'32$, luego el número total de palabras con 4 letras que pueden formarse es $2^{R\cdot 4} = (2^R)^4 = 5^4 = 625$, mientras que el número de palabras con sentido, si r está entre 1'2 y 1'5 estará comprendido entre $2^{1'2\cdot 4}$ y $2^{1'5\cdot 4}$, es decir, entre $27'8$ y 64. En este caso, el número de palabras con sentido es 45, y éstas son:

> aeta, asas, asea, asee, aseo, ases, asta, atea, atas, ates, ateo, atoa, atoe, atoo, osas, oses, osos, oste, otea, otee, oteo, easo, esas, eses, esos, esta, este esto, etas, tasa, tase, taso, teas, tesa, tese, teso, teta, seas, seso, seta, seto, sosa, sota, sote, soto.

Se define entonces la redundancia del lenguaje como la diferencia entre el índice absoluto y el índice del lenguaje. En el caso del español, la redundancia está comprendida entre $4'75 - 1'5$ y $4'75 - 1'2$, es decir, entre $3'25$ y $3'55$.
Significa esto que en un texto español, entre un $68'45\,\%$ y un $74'76\,\%$ es redundante (estas cantidades se han obtenido como $\frac{3'25}{\log_2(27)} \cdot 100$ y $\frac{3'55}{\log_2(27)} \cdot 100$).
Como hemos dicho, esta redundancia ha sido la que nos ha servido para romper algunos criptosistemas. A la hora de cifrar un mensaje, se trata de desvirtuar de alguna forma todas las características propias del lenguaje que se está empleando.
Para esto se utilizan dos procesos, como son el de *difusión* y *confusión*.
Difusión.
Con esto se pretende lo siguiente: uniformizar las frecuencias de las letras en el mensaje cifrado. De esta forma evitamos ciertos tipos de análisis del criptosistema. Esto normalmente se va a realizar mediante el manejo de (sub)bloques del texto.
Confusión.
Aquí destacamos el siguiente aspecto: desaparición de patrones propios del idioma, por ejemplo grupos de letras. Normalmente se realizar mediante una permutación de las letras del texto.

Parte II

Cifrado en flujo

CRIPTOGRAFÍA IV

Cifrados en flujo

Los métodos criptográficos actuales podemos clasificarlos como sigue:

$$\text{Métodos cifrado} \begin{cases} \text{Flujo}: \ A5,\ RC4 \\ \text{Bloque} \begin{cases} \text{Clave secreta: } DES,\ TDES,\ IDEA,\ AES \\ \text{Clave pública:} \begin{cases} \text{Exponenciación: } RSA,\ ElGammal \\ \text{suma/producto: Curvas elípticas} \end{cases} \end{cases} \end{cases}$$

Cada uno de éstos tiene un uso diferente. Así, los sistemas de flujo son más empleados en telefonía movil, o en WLAM, los de clave secreta en una sesión de internet o cifrado local, mientras que los de clave pública, por ejemplo, en intercambio de claves o en la firma digital. Nos proponemos en este capítulo estudiar los cifrados por flujo.

Éstos se basan en el concepto de cifra ideado por Vernam. En este sistema se creaba una sucesión de bits aleatoria del mismo tamaño del mensaje, y se realizaba una suma lógica $\oplus$ bit a bit, entre la sucesión de bits del mensaje y la sucesión aleatoria de bits. Desde el punto de vista de la teoría de la información estudiada en el capítulo anterior, este es un sistema criptográfico perfecto.

Una vez se había utilizado la clave, ésta se destruía (lo que se conoce como sistema one-time pad).

Sin embargo, el sistema de Vernam carece de utilidad práctica. Si tenemos que enviar a nuestro receptor la clave aleatoria que hemos generado, y esta es del mismo tamaño que el mensaje, entonces ¿por qué no enviar el propio mensaje en claro?. Necesitaríamos un canal seguro para enviar la clave, y ese canal nos serviría entonces para enviar el mensaje.

Los cifrados por flujo se basan en sustituir las sucesiones aleatorias por *sucesiones criptográficamente aleatorias.* Éstas son sucesiones de bits generadas mediante un algoritmo a partir de una semilla, de forma que la sucesión resultante supere ciertos test de aleatoriedad (es lo que se conoce como una sucesión pseudoaleatoria).

A partir de una semilla de n bits se pueden generar secuencias de periodo 2^n.

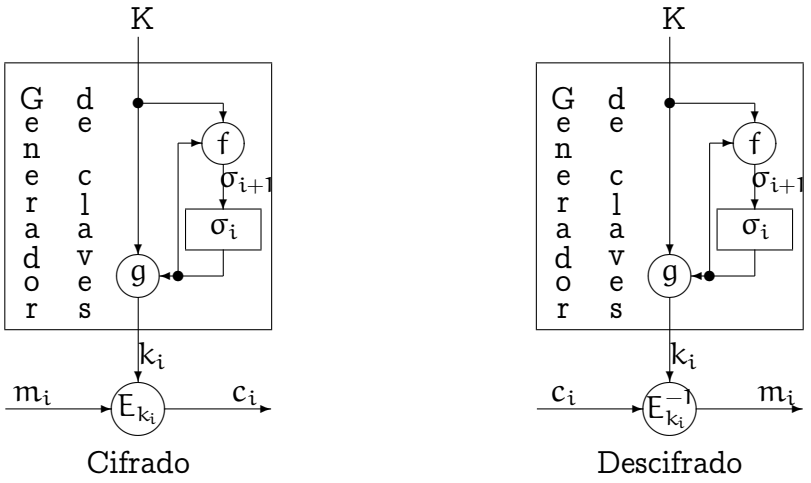
Por tanto, la técnica de cifrado en flujo requiere leer el mensaje en claro bit a bit, y generar una secuencia de bits, de periodo muy alto, que ya no será aleatoria, sino determinista, pero satisfará determinadas propiedades pseudoaleatorias. Luego se realiza una operación de cifra con el mensaje, que normalmente consistirá en la suma lógica (XOR)

En general podemos clasificar los cifrados por flujo en dos grandes grupos: *síncronos* y *asíncronos*.

. . . . . 0.1

Cifrados por flujo síncronos

En estos, la sucesión criptográfica se genera independientemente del texto en claro y del criptograma.
Consistiría entonces en un generador de claves, que a partir de una clave de inicialización K (semilla) genera una sucesión de bits k_i.
Normalmente, el generador de claves es un dispositivo que va pasando por sucesivos estados σ_i. Cada estado depende del estado anterior, y posiblemente de la clave K. A su vez, cada estado da lugar a un bit k_i de la secuencia, que unido al bit m_i del mensaje da lugar al bit c_i del criptograma. Tendríamos entonces:
A partir de una semilla K se obtiene el estado inicial del dispositivo σ_0. A partir de un estado σ_i, y quizá dependiendo de la clave K se obtiene el estado σ_{i+1}. Esta dependencia podemos representarla mediante una función f, que se denominaría *función estado siguiente*.
A su vez, el estado σ_i da lugar a un bit k_i que depende de dicho estado y eventualmente de la clave K. Representaremos esto mediante $k_i = g(\sigma_i, K)$
Por último, tendríamos una función de cifrado E_{k_i}, que a partir de un bit de texto en claro m_i produce el bit $c_i = E_{k_i}(m_i)$. Normalmente, la función E_{k_i} suele ser la función $m_i \mapsto m_i \oplus k_i$.
Para descifrar se procede de igual forma. La generación de la clave se hace exactamente igual (pues no depende del texto plano y el criptograma), y se descifra mediante la función $E_{k_i}^{-1}$ (si E_{k_i} es la que hemos especificado previamente, entonces $E_{k_i}^{-1} = E_{k_i}$).



Es decir, el generador de la secuencia pseudoaleatoria va pasando por distintos estados $\sigma_0, \sigma_1, \cdots \sigma_i, \sigma_{i+1}, \cdots$ y en función de cada uno de esos estados se obtiene un bit de salida que es el que utilizamos para cifrar.
Para descifrar, se sigue el mismo proceso. Para obtener la misma secuencia de bits es necesario que el estado inicial σ_0, y la clave K sean las mismas. Esto último significa que los generadores están sincronizados, de ahí el nombre de *cifrado en flujo síncrono*.

Veamos algún ejemplo.
Supongamos que nuestro generador pseudoaleatorio consta de 4 celdas, cada una de las cuales contiene un bit. Denotemos el contenido de cada celda como $c(1), c(2), c(3)$ y $c(4)$.
Un estado σ_i será por tanto un valor concreto de los contenidos de las celdas. Por ejemplo, un estado puede ser $(1,0,1,1)$, lo que significa que $c(1) = 1$, $c(2) = 0$, $c(3) = 1$ y $c(4) = 1$.
La función f sería $f((c(1), c(2), c(3), c(4)) = (c(2) + c(4), c(1), c(2), c(3))$, es decir, f desplaza las tres primeras celdas una posición, y coloca en la primera la suma de la segunda y la cuarta (en este caso no depende de la clave).
La función g lo que hace es sumar los bits $c(2)$ y $c(4)$, es decir, $g((c(1), c(2), c(3), c(4)) = c(2) + c(4)$
Por último, la función de cifrado es la función XOR.
El estado inicial σ_0 es la semilla.
Si tenemos el mensaje $M = 10110101$, y partimos del estado inicial $\sigma_0 = (0,1,1,1)$ obtendríamos:

i		$c(1)$	$c(2)$	$c(3)$	$c(4)$	k_i	m_i	c_i
0	σ_0	0	1	1	1	0	1	1
1	σ_1	0	0	1	1	1	0	1
2	σ_2	1	0	0	1	1	1	0
3	σ_3	1	1	0	0	1	1	0
4	σ_4	1	1	1	0	1	0	1
5	σ_5	1	1	1	1	0	1	1
6	σ_6	0	1	1	1	0	0	0
7	σ_7	0	0	1	1	1	1	0

Y por tanto, el mensaje cifrado es $C = 11001100$.
Para poder descifrar, necesitamos partir del mismo estado inicial $(0,1,1,1)$, y a partir de ahí generar la secuencia k_i y sumársela al criptograma.
Entre las principales características de estos cifrados debemos destacar:

Sincronización obligada El emisor y el receptor deben utilizar la misma sucesión en la misma posición. Si algún bit se pierde o se inserta en la transmisión, el descifrado falla, y sólo puede recuperarse mediante técnicas adicionales de resincronización.

No propagación de errores Si un bit se altera en la transmisión dicha alteración no afecta al resto del proceso de descifrado.

Ataques activos Las características anteriores hacen que estos criptosistemas sean fuertes ante la inserción y borrado de información, pero débiles ante la alteración de la misma.

Por ejemplo, si la función de cifrado es la suma lógica, y el atacante conoce un trozo del texto en claro m, entonces puede sustituirlo por un nuevo mensaje m' sin que el destinatario lo note. Lo único que tiene que hacer es sustituir el criptograma enviado c por $c' = c \oplus m \oplus m'$.

Supongamos que el destinatario recibe el criptograma c' y se dispone a descifrarlo. Para ello usa la secuencia clave k (si k es la secuencia de cifrado, entonces $c = m \oplus k$). Entonces calcularía:

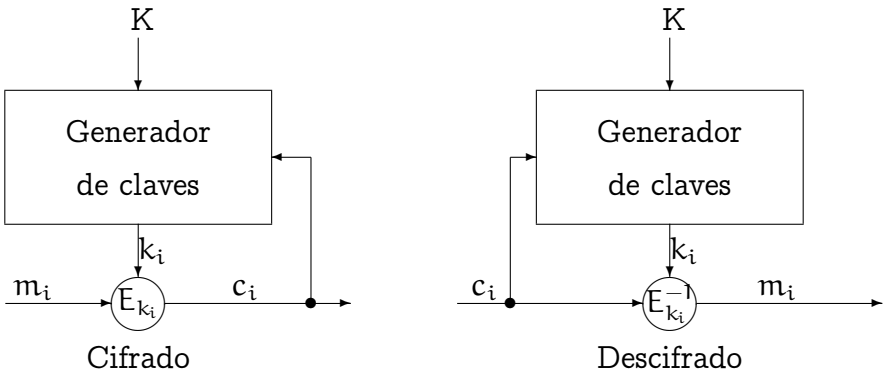
$$c' \oplus k = c \oplus m \oplus m' \oplus k = c \oplus m \oplus k \oplus m' = c \oplus c \oplus m' = m'$$

. . . . . 0.2

Cifrados por flujo asíncronos o autosincronizantes

La diferencia fundamental con los cifrados síncronos es que aquí la secuencia que se genera depende del mensaje.

Un esquema de este tipo de cifradores podría ser



Vamos a ver un ejemplo.

Al igual que antes, nuestro generador pseudoaleatorio consta de 4 celdas, cada una de las cuales contiene un bit, y el contenido de cada celda lo denotaremos como $c(1), c(2), c(3)$ y $c(4)$.

Cada valor concreto de las celdas $(c(1), c(2), c(3), c(4))$ es un estado del generador.

A partir de un estado, se obtiene el bit de la clave como $c(1) + c(4)$.

La función de cifrado es una suma lógica del bit del mensaje y el bit de la clave.

El nuevo estado se obtiene desplazando las tres primeras celdas una posición, y colocando en la primera celda el bit cifrado.

El estado inicial σ_0 es la semilla.

Para descifrar, se parte del estado inicial σ_0. El bit de la clave se obtiene, como antes realizando la suma $c(1) + c(4)$, y este se suma al bit correspondiente del criptograma, obteniéndose así el bit del texto plano.

El siguiente estado se obtendría desplazando las tres primeras celdas una posición, y colocando en la primera celda el bit del criptograma.

Supongamos que tenemos un mensaje $M = 101011001011$, y tomamos como estado inicial $\sigma_0 = (0, 1, 1, 1)$. Entonces:

i		$c(1)$	$c(2)$	$c(3)$	$c(4)$	k_i	m_i	c_i
0	σ_0	0	1	1	1	1	1	0
1	σ_1	0	0	1	1	1	0	1
2	σ_2	1	0	0	1	0	1	1
3	σ_3	1	1	0	0	1	0	1
4	σ_4	1	1	1	0	1	1	0
5	σ_5	0	1	1	1	1	1	0
6	σ_6	0	0	1	1	1	0	1
7	σ_7	1	0	0	1	0	0	0
8	σ_8	0	1	0	0	0	1	1
9	σ_9	1	0	1	0	1	0	1
10	σ_{10}	1	1	0	1	0	1	1
11	σ_{11}	1	1	1	0	1	1	0

Y por tanto, el mensaje cifrado es $C = 011100101110$.

Para descifrar, tendríamos:

i	c_{i-1}		$c(1)$	$c(2)$	$c(3)$	$c(4)$	k_i	c_i	m_i
0		σ_0	0	1	1	1	1	0	1
1	0	σ_1	0	0	1	1	1	1	0
2	1	σ_2	1	0	0	1	0	1	1
3	1	σ_3	1	1	0	0	1	1	0
4	1	σ_4	1	1	1	0	1	0	1
5	0	σ_5	0	1	1	1	1	0	1
6	0	σ_6	0	0	1	1	1	1	0
7	1	σ_7	1	0	0	1	0	0	0
8	0	σ_8	0	1	0	0	0	1	1
9	1	σ_9	1	0	1	0	1	1	0
10	1	σ_{10}	1	1	0	1	0	1	1
11	1	σ_{11}	1	1	1	0	1	0	1

Supongamos que los generadores de claves no estuvieran sincronizados. Por ejemplo, el receptor tuviera como estado inicial $\sigma_0 = (1, 0, 1, 0)$. Vamos a ver como afecta al proceso de descifrado (pondremos en negrita aquellos bits en los que se produce error)

i	c_{i-1}		$c(1)$	$c(2)$	$c(3)$	$c(4)$	k_i	c_i	m_i
0		σ_0	**1**	**0**	1	**0**	1	0	1
1	0	σ_1	0	**1**	**0**	1	1	1	0
2	1	σ_2	1	0	**1**	**0**	**1**	1	**0**
3	1	σ_3	1	1	0	**1**	**0**	1	**1**
4	1	σ_4	1	1	1	0	1	0	1
5	0	σ_5	0	1	1	1	1	0	1
6	0	σ_6	0	0	1	1	1	1	0
7	1	σ_7	1	0	0	1	0	0	0
8	0	σ_8	0	1	0	0	0	1	1
9	1	σ_9	1	0	1	0	1	1	0
10	1	σ_{10}	1	1	0	1	0	1	1
11	1	σ_{11}	1	1	1	0	1	0	1

Y vemos como a partir del quinto estado, los dos generadores se han sincronizado. El error afecta entonces únicamente a dos bits del mensaje. Por tanto, a la hora de transmitir un mensaje, el emisor puede añadir unos bits adicionales al inicio. De esta forma, el receptor no tendrá porqué sincronizar su generador, pues éste se sincroniza sólo. Una vez descifrado el criptograma, bastará con eliminar los primeros bits para obtener el texto en claro.
Las características a destacar son:

Autosincronización Si algunos dígitos se pierden o se insertan es posible volver a sincronizar el proceso de descifrado debido a que éste depende sólo de un número fijo de bits precedentes. Sólo se perderían algunos bits del texto en claro transmitido.

Propagación de errores limitada Un error (del tipo que sea) en la transmisión de un bit afecta exclusivamente a los siguientes t bits, por lo que la transmisión puede recuperarse a partir de ellos.

Ataques activos Comparados con los cifrados por flujo síncronos es más fácil detectar la modificaión de bits en la transmisión, ya que dicha modificación afecta a más bits. Sin embargo es más difícil detectar inserción o borrado de bits.

Difusión de la estructura del texto en claro Cada dígito del texto en claro influye en todo el criptograma generado a partir de él. Si el texto en claro presenta una estructura definida, esta estructura se dispersa en todo el criptograma, al contrario que en los cifrados por flujo síncronos en los que dicha estructura permanece localizada.

1 Generación de secuencias aleatorias

Hemos visto que para el cifrado en flujo, un apartado importante es la generación de secuencias aleatorias. Para obtener una secuencia aleatoria de n bits, bastaría con que una persona

lanzara una moneda n veces al aire, y anotara el resultado. Pero no sería práctico para la obtención de secuencias largas de bits. Además, dicha secuencia no seria reproducible.
Existen valores obtenidos del hardward de la computadora que suelen proporcionar bits de aleatoriedad. Si el ordenador dispone de un reloj de alta precisión, la lectura del estado interno de dicho reloj puede proporcionar un resultado prácticamente impredecible, por lo que se podría emplear para obtener valores aleatorios. También pueden servir tarjetas digitalizadoras de sonido o vídeo, con la sensibilidad suficiente para captar el ruido térmico. O también si somos capaces de medir las fluctuaciones en la velocidad de giro de unidades de disco, debidas, por ejemplo, a turbulencias en el aire, pueden proporcionarnos una buena muestra de valores aleatorios.
También pueden emplearse funciones de mezcla, si, por ejemplo, no se dispone de una fuente fiable de bits aleatorios, pero se dispone de varias fuentes menos fiables. Los algoritmos de cifrado simétrico (tipo DES o AES), o las funciones resumen son un buen ejemplo de funciones de mezcla.
Sin embargo, estas secuencias no pueden reproducirse, por tanto, a menos que se consiga transmitir, no son válidas para el cifrado en flujo. Como ya comentamos previamente, lo que se hace es construir secuencias pseudoaleatorias. Éstas se consiguen a partir de un valor inicial o semilla, y un algoritmo determinista. Se obtiene así una secuencia periódica, que debe satisfacier ciertos test de aleatoriedad.

2 Secuencias pseudoaleatorias

Uno de los test de aleatoriedad es el propuesto por Golomb, en 1982. Según esta propuesta, una secuencia pseudoaleatoria debe satisfacer los siguienes postulados:

1. En todo período, la diferencia entre el número de unos y el número de ceros debe ser a lo sumo uno.

2. En un periodo, el número de rachas de longitud 1 debe ser el doble al número de rachas de longitud 2, y este a su vez, el doble de rachas de longitud 3, etc.

3. La *distancia de Hamming* entre dos secuencias diferentes, obtenidas mediante desplazamientos circulares de un periodo, debe ser constante.

Aclaremos brevemente que significan estos postulados.
El primer postulado está claro. Basta considerar un periodo y contar el número de ceros y de unos. Por ejemplo, la secuencia de periodo 111011011000100 satisface el primer postulado de Golomb, pues está formada por 8 "unos" y 7 "ceros". Sin embargo, la secuencia 0101110010010001 no satisface este postulado, pues está formada por 7 "unos" y 9 "ceros".
Para el segundo postulado, expliquemos en primer lugar qué es una racha. Una racha es un grupo de bits iguales (podría ser de un solo bit) entre dos dígitos distintos. Por ejemplo, en la siguiente secuencia 000111101011001 tenemos las siguientes rachas:

- De longitud 1: 0 0 0 1 1 1 1 **0** **1** **0** 1 1 0 0 **1**. La primera es un 0 entre dos unos, la segunda un 1 entre dos ceros, la tercera un 0 entre dos unos, y la cuarta un 1 entre dos ceros (al ser esta secuencia un periodo, el siguiente dígito sería 0).
- De longitud 2: 0 0 0 1 1 1 1 0 1 0**1 1 0 0**1. Es decir, un grupo 11 entre dos ceros, y un grupo 00 entre dos unos.
- De longitud 3: **000**111101011001.
- De longitud 4: 000**1111**01011001.

Vemos que hay cuatro rachas de longitud 1, dos rachas de longitud 2, una racha de longitud 3 y una de longitud 4. Esta secuencia satisface por tanto el segundo postulado de Golomb.
Para el tercero, la distancia de Hamming entre dos secuencias es el número de bits diferentes. Por ejemplo, partimos de la secuencia anterior, y la desplazamos una posición a la derecha. Vamos a contar el número de bits diferentes entre cada una de las dos secuencias:

0 0 0 1 1 1 1 0 1 0 1 1 0 0 1	Vemos que hay 8 bits diferentes,
1 0 0 **0** 1 1 1 **1** **0** **1** **0** 1 **1** 0 **0**	luego la distancia de Hamming es 8

Por ejemplo, si consideramos la secuencia 0011101, las distintas distancias de Hamming entre las secuencias que se obtienen desplazando cíclicamente la secuencia original están calculadas en la siguiente tabla:

	0	0	1	1	1	0	1	distancia
$k=1$	**1**	0	**0**	1	1	**1**	**0**	4
$k=2$	0	**1**	**0**	**0**	1	**1**	1	4
$k=3$	**1**	0	1	**0**	**0**	**1**	1	4
$k=4$	**1**	**1**	**0**	1	**0**	0	1	4
$k=5$	**1**	**1**	1	**0**	1	0	**0**	4
$k=6$	0	**1**	1	1	**0**	**1**	**0**	4
$k=7$	0	0	1	1	1	0	1	0

La distancia se calcula siempre respecto a la secuencia original. Como vemos, la distancia de Hamming es en todos los casos 4, por tanto es constante, y esta secuencia satisface el tercer postulado de Golomb.
Si consideramos la siguiente secuencia 01110100, calculamos las distancias de Hamming y nos da:

	0	1	1	1	0	1	0	0	distancia
$k=1$	0	**0**	1	1	**1**	**0**	**1**	0	4
$k=2$	0	**0**	**0**	1	**1**	1	0	**1**	4
$k=3$	**1**	**0**	**0**	**0**	**1**	1	**1**	0	6
$k=4$	0	1	**0**	**0**	0	1	**1**	**1**	4
$k=5$	**1**	**0**	1	**0**	0	**0**	**1**	**1**	6
$k=6$	**1**	1	**0**	1	0	**0**	0	**1**	4
$k=7$	**1**	1	1	**0**	**1**	**0**	0	0	4
$k=8$	0	1	1	1	0	1	0	0	0

Y vemos aquí como la distancia de Hamming no es constante, luego esta secuencia no satisface el tercer postulado de Golomb.

. . . . . 2.1

Generadores de secuencias pseudoaleatorias

En esta sección vamos a estudiar distintos generadores de secuencias pseudoaleatorias. Vamos a comenzar por las más sencillas, e iremos viendo los inconvenientes que presentan, lo que nos obligará a buscar otro tipo de generadores.

Generador de congruencia lineal

Para este generador, necesitamos tres números enteros a, b, n. Entonces, a partir de un valor inicial x_0 se obtiene la secuencia cifrante $x_{i+1} = ax_i + b(\text{mód } n)$. Por ejemplo, si tomamos $a = 4$, $b = 2$ y $n = 7$, tomando como semilla $x_0 = 2$, nos sale la secuencia:

$$2\ 1\ 6\ 8\ 7\ 3\ 5\ 4\ 0\ 2\cdots$$

En estos casos, se suele emplear como n una potencia de 2, así podemos tomar la representación binaria de cada elemento de la secuencia. Así, si tomamos $a = 5$, $b = 1$ y $n == 2^4 = 16$, y como semilla $x_0 = 10$, tenemos la secuencia:

$$10\ 3\ 0\ 1\ 6\ 15\ 12\ 13\ 2\ 11\ 8\ 9\ 14\ 7\ 5\ 10\cdots$$

que nos daría la sucesión de bits

$$10100011000000010110111111001101001010111000100111100111010 1$$

y a partir de aquí se repiten.
Sin embargo, si analizamos distintos casos, vemos que las secuencias que nos salen pueden no ser muy buenas. Veamos los siguientes ejemplos:

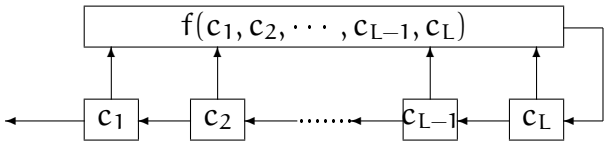
a	b	n	x_0	secuencia
5	2	16	10	$10\ 4\ 6\ 0\ 2\ 12\ 14\ 8\ 10\cdots$
5	2	16	7	$7\ 5\ 11\ 9\ 15\ 13\ 3\ 1\ 7\cdots$
4	1	16	7	$7\ 13\ 5\ 5\cdots$

Vemos entonces que estas secuencias no son criptográficamente interesantes.

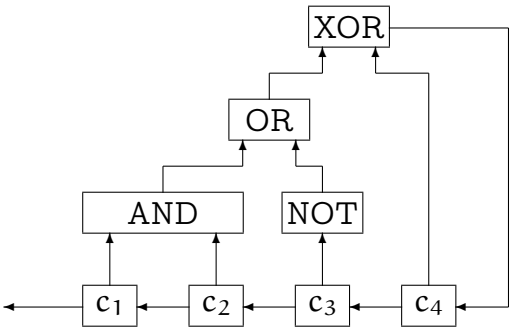
Registros de desplazamiento

En lugar entonces del generador de congruencia lineal se suele emplear los registros de desplazamiento, que ya hemos visto en algunos ejemplos anteriores. Un registro de desplazamiento es una memoria, de L celdas, cuyo contenido se desplaza con los pulsos de un reloj de control. Al mismo tiempo, el contenido de las celdas se combina, bien mediante operaciones lineales o no lineales, y eventualmente, el resultado puede realimentar a la última celda

de la memoria. Por tanto, dentro de los registros de desplazamiento (SR - Shift Register -) se pueden destacar los registros de desplazamiento con retroalimentación (FSR - Feedback Shift Register), y dentro de éstos, los registros de desplazamiento con retroalimentación lineal (LFSR) o los registros de desplazamiento con retroalimentación no lineal (NLFSR)
Vamos a centrarnos en los registros de desplazamiento retroalimentados. Como hemos dicho, están constituidos por L celdas $c_1, c_2, \cdots, c_L$, y una función que realimenta la última celda, a la vez que las otras se desplazan, mientras que devuelve el bit que se encuentra en la primera celda. El esquema de funcionamiento podría ser:



Tras el pulso de reloj, el generador devuelve el valor almacenado en c_1, a la vez que desplaza a la izquierda los almacenados en $c_2, \cdots c_{L-1}, c_L$. En la última celda se almacena el valor resultado de aplicar la función f a los valores $c_1, c_2, \cdots, c_{L-1}, c_L$.
Un ejemplo de Registro de desplazamiento con retroalimentación, podría ser:



Así, por ejemplo, si partimos de 1011, entonces tras un pulso de reloj, nos devuelve 1, las celdas primera, segunda y tercera toman los valores 011, mientras que la cuarta toma el valor 1, resultado de hacer las operaciones que se indican 1 AND $0 = 0$, NOT $1 = 0$, 0 OR $0 = 0$ y 0 XOR $1 = 1$. Por tanto, el generador nos devuelve el valor 1 y se sitúa en el estado 0111.
Siguiendo el proceso, obtenemos la secuencia:

$$101111010010\cdots$$

Podemos ver como el valor que retroalimenta a la cuarta celda es $1 + c_1c_2c_3 + c_3 + c_4$.
Esta es una función no lineal, luego se trata de un registro de desplazamiento con retroalimentación no lineal.
Vamos a pasar a estudiar con más detalle los Registros de desplazamiento con retroalimentación lineal (LFRS). Éstos son muy utilizados por varias razones:

1. fáciles de implementar en hardware,

2. producen sucesiones de período grande,
3. producen sucesiones con buenas propiedades estadísticas,
4. pueden ser analizados mediante técnicas algebraicas.

En estos generadores, supongamos que las celdas $c_1, c_2, \cdots, c_L$ tienen almecenados los bits $s_j, s_{j+1}, \cdots, s_{j+L-1}$. En tal caso, el bit siguiente se calcula mediante una función lineal de los bits precedentes, es decir, s_{j+L} es una función lineal de $s_j, s_{j+1}, \cdots, s_{j+L-1}$. Adopta por tanto la forma

$$s_{j+L} = a_1 \cdot s_{j+L-1} + a_2 \cdot s_{j+L-2} + \cdots + a_L \cdot s_j$$

donde a_i vale 0 o 1 (las operaciones están realizadas en $\mathbb{Z}_2$), o lo que es lo mismo,

$$1 \cdot s_{j+L} + a_1 s_{j+L-1} + a_2 s_{j+L-2} + \cdots + a_L s_j = 0$$

A partir de esto se define el polinomio de conexión $c(D)$ (que nos da las operaciones entre celdas) como

$$c(D) = 1 + a_1 D + a_2 D^2 + \cdots a_L D^L$$

Por tanto, un LFSR depende de dos parámetros, la longitud L y su polinomio de conexión $c(D) = 1 + a_1 D + a_2 D^2 + \cdots + a_L D^L \in \mathbb{Z}_2[D]$. Nótese que $c(0) = 1$.
Dado un LFSR de parámetros $(c(D), L)$, entonces a partir de un estado inicial $s_0, s_1, \cdots s_{L-1}$, se define la sucesión de bits s_n de forma recursiva como

$$s_{j+L} = a_1 \cdot s_{j+L-1} + a_2 \cdot s_{j+L-2} + \cdots + a_L \cdot s_j$$

Las propiedades algebraicas del polinomio $c(D)$ nos van a determinar las características de la secuencia s_n.
Veamos algunos ejemplos:
Sea $c(D) = D^4 + D^2 + 1$, y tomamos, por ejemplo $s_0 = 1$, $s_1 = 0$, $s_2 = 0$ y $s_3 = 1$. En este caso, $s_{j+4} = s_{j+2} + s_j$. Así, $s_4 = s_2 + s_0 = 1$, $s_5 = s_3 + s_1 = 1$, y la sucesión resultante es

$$100111100111100111100111\cdots$$

Y vemos que la sucesión tiene periodo de longitud 6 (100111) De hecho, la sucesión es la superposición de dos sucesiones de periodo 3, (101 y 011) que se van intercalando en las posiciones impares y pares.
Si la semilla fuera 1101, tendríamos la secuencia

$$110110110110$$

que tiene periodo de longitud 3.
Sea ahora $c(D) = D^4 + D^3 + D^2 + D + 1$, y $s_0 = 1$, $s_1 = 1$, $s_2 = 0$ y $s_3 = 1$. La sucesión que obtenemos en este caso es

$$1101111011110111101111\cdots$$

Y lo que obtenemos es una secuencia de periodo 5.
Sea ahora $c(D) = D^4 + D + 1$, y $s_0 = 1$, $s_1 = 0$, $s_2 = 0$, $s_3 = 1$. Para obtener el resto de la sucesión tenemos la expresión $s_{j+4} = s_{j+3} + s_j$. Nos queda entonces:

$$1001000111101011001000111101011001000\cdots$$

Y ahora el periodo es de longitud 15 (100100011110101). Nótese que en este caso, el generador ha pasado por los 15 estados posibles (salvo el 0000, pues al ser lineal, si tuviera ese estado, todos los bits siguentes de la sucesión serían cero). Estos 15 estados son:

$$1001 \quad 0010 \quad 0100 \quad 1000 \quad 0001 \quad 0011 \quad 0111 \quad 1111$$
$$1110 \quad 1101 \quad 1010 \quad 0101 \quad 1011 \quad 0110 \quad 1100$$

Por tanto, el periodo en este caso es máximo (15). Además, se puede comprobar que satisface los 3 postulados de Golomb. Tenemos tres casos en función del polinomio:

1. El polinomio es reducible en $\mathbb{Z}_2[x]$, como por ejemplo $D^4 + D^2 + 1 = (D^2 + D + 1)^2$. En tal caso, el periodo es menor que el periodo máximo $2^L - 1$, y depende de la semilla, como hemos visto antes. Las propiedades de la sucesión dependen de cómo se factorice el polinomio, pero no son criptográficamente interesantes.

2. El polinomio es irreducible, pero no es primitivo. Vamos a explicar brevemente que significa esto de ser primitivo, aunque lo veremos más en profundidad al estudiar el criptosistema AES.

   Al ser $c(D)$ irreducible, si hacemos el cociente $\mathbb{Z}_2[D]/(c(D))$ es un cuerpo (de la misma forma que al hacer $\mathbb{Z}/p\mathbb{Z}$ con p primo nos sale cuerpo). En tal caso, al hacer todas las potencias de un elemento no nulo pueden salirnos todos los elementos del cuerpo (salvo el cero) o no. En el primer caso, el elemento se dice primitivo, en el segundo no. Si tomamos el elemento $\alpha = [D]$, puede ocurrir que sea primitivo o que no lo sea. Cuando lo sea, entonces diremos que $c(D)$ es primitivo. En caso contrario diremos que el polinomio no es primitivo.

   Si tomamos $c(D) = D^4 + D^3 + D^2 + D + 1$, entonces al hacer las potencias de $\alpha = [D]$ obtenemos únicamente 5 potencias distintas ($\alpha^5 = 1$), por tanto $c(D)$ no es primitivo. Si lo hacemos con $c(D) = D^4 + D + 1$, entonces la primera potencia de α que nos sale igual a 1 es α^{15}, por tanto, haciendo las potencias de α obtenemos los 15 elementos distintos de cero de $\mathbb{Z}_2[D]/(c(D))$, luego este polinomio sí es primitivo.

   Volviendo a donde estábamos antes, si el polinomio es irreducible, pero no primitivo, el periodo no depende de la semilla, pero es un divisor del periodo máximo (que es $2^L - 1$). En el ejemplo que hemos visto nos ha salido que el periodo es 5, que es un divisor de $2^4 - 1 = 15$. Las sucesiones así obtenidas no son criptográficamente interesantes.

3. El polinomio es irreducible y primitivo. En este caso, la sucesión tiene periodo $2^L - 1$, y el periodo satisface los tres postulados de Golomb.

Este último caso es el que genera sucesiones criptográficamente interesantes. La sucesión generada por un LFSR de parámetros $c(D), L$, donde $c(D)$ es un polinomio primitivo tiene la siguientes características:

- El periodo es $2^L - 1$, sea cual sea la semilla (salvo la $00\cdots0$)
- El número de unos es 2^{L-1}, y el número de ceros es $2^{L-1} - 1$.
- El número de rachas viene dado en la siguiente tabla:

Longitud de la racha	Rachas de ceros	Rachas de unos	Rachas
1	2^{L-3}	2^{L-3}	2^{L-2}
2	2^{L-4}	2^{L-4}	2^{L-3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
L-2	1	1	2
L-1	1	0	1
L	0	1	1

- La distancia de Hamming entre dos periodos cualesquiera es 2^{L-1}.

Vemos por tanto, que con esta elección del polinomio para un LFSR obtenemos una buena secuencia pseudoaleatoria. Además, podemos hacer el periodo lo suficientemente grande sin más que aumentar el grado del polinomio. Así, con grado 10 obtenemos un periodo de 1023 bits, para grado 20 el periodo es de 1048575 bits, y para grado 30 nos sale de 1073741823. Parece ser que con esto hemos resuelto el problema de generación de secuencias pseudoaleatorias, pero vamos a ver que no todo es tan sencillo.
El problema viene dado por la forma lineal. Esto permite que en una secuencia de un periodo $2^L - 1$, con el conocimiento de 2L bits, se pueda determinar toda la secuencia. Veamos un ejemplo. Consideramos la secuencia generada anteriormente a partir del polinomio D^4+D+1, que tiene periodo 15. Supongamos que un atacante consigue hacerse con 8 bits consecutivos de la secuencia, por ejemplo, $s_0s_1s_2s_3s_4s_5s_6s_7 = 10010001$. Entonces, puede plantear el sistema de ecuaciones con coeficientes en $\mathbb{Z}_2$

$$\begin{array}{l} s_4 = s_3a_1 + s_2a_2 + s_1a_3 + s_0a_4 \\ s_5 = s_4a_1 + s_3a_2 + s_2a_3 + s_1a_4 \\ s_6 = s_5a_1 + s_4a_2 + s_3a_3 + s_2a_4 \\ s_7 = s_6a_1 + s_5a_2 + s_4a_3 + s_3a_4 \end{array} \quad \text{es decir} \quad \begin{array}{rcccccc} 0 &=& a_1 & & & + & a_4 \\ 0 &=& & a_2 & & & \\ 0 &=& & & a_3 & & \\ 1 &=& & & & & a_4 \end{array}$$

que tiene como solución $a_1 = a_4 = 1$, $a_2 = a_3 = 0$, lo que nos da el polinomio $D^4 + D + 1$.
En general, el conocimiento de 2L bits consecutivos $s_js_{j+1}\cdots s_{j+2L-2}s_{j+2L-1}$ nos permite plantear el sistema:

$$
\begin{array}{lcccccccc}
s_{j+L} & = & s_{j+L-1}a_1 & + & s_{j+L-2}a_2 & + & \cdots & + & s_j a_L \\
s_{j+L+1} & = & s_{j+L}a_1 & + & s_{j+L-1}a_2 & + & \cdots & + & s_{j+1} a_L \\
\multicolumn{9}{c}{\dotfill} \\
s_{j+2L-1} & = & s_{j+2L-2}a_1 & + & s_{j+2L-3}a_2 & + & \cdots & + & s_{j+L-1} a_L
\end{array}
$$

que tiene L ecuaciones y L incógnitas, y cuya solución nos da el polinomio $c(D)$.
Nos aparece entonces un concepto, que es el de la complejidad lineal, que es la longitud mínima del LFSR capaz de producirla. Nótese que toda secuencia pseudoaleatoria periódica puede ser generada por un generador lineal (de longitud igual al periodo). El problema de los LFSR, es que, aunque generan secuencias con buenas propiedades estadísticas de aleatoriedad, tienen una complejidad lineal muy baja en relación con el periodo. Lo que hay que hacer es tratar de aumentar la complejidad lineal.
Para solucionar este problema, se han propuesto varias alternativas:

1. uso de NLFSR, es decir, registros de desplazamiento retroalimentados no lineales. Ya vimos un ejemplo de esto.

2. uso de una combinación de las salidas de varios LFSR

3. uso de la salida de al menos un LFSR para controlar el reloj de los demás LFSRs.

Vamos a ver algunos ejemplos de las soluciones anteriores.

Suma de secuencias

Dadas dos (o más) secuencias, s_n y s'_n, generadas por un LFSR, podemos obtener una nueva secuencia sumando ambas, es decir, obtendríamos la secuencia $s_n + s'_n$. Si ambas secuencias se han obtenido con polinomios primitivos $c(D)$ y $c'(D)$, y sus periodos son $2^L - 1$ y $2^{L'-1}$, entonces, la secuencia suma tiene periodo $mcm(2^L-1, 2^{L'}-1)$. La complejidad lineal en este caso es la suma de las complejidades lineales de ambos generadores.
Veamos un ejemplo sencillo. Sean los LFSR de parámetros $(D^2+D+1,\ 2)$ y $(D^3+D+1,\ 3)$, y consideramos las secuencias que generan a partir de las semillas 11 y 111 respectivamente:

$$110110110110110110110\cdots \qquad 111010011101001110100\cdots$$

que vemos tienen periodos 3 y 7 respectivamente. Realizamos la suma de ambos

$$
\begin{array}{c}
110110110110110110110110110110110110110110 \\
111010011101001110100111010011101001110100 \\
\hline
001100101011111000010001100101011111000010
\end{array}
$$

que como vemos tiene periodo 21. Podemos comprobar que esta secuencia podría haber sido generada por un LFSR de parámetros $D^5 + D^4 + 1, 5$.

Multiplicación de secuencias

Lo que hacemos aquí es multiplicar las sucesiones de bits que nos resultan. El periodo vuelve a ser en este caso $mcm(2^L-1, 2^{L'}-1)$, mientras que la complejidad lineal es $L \cdot L'$. En el ejemplo anterior, tendríamos:

$$\begin{array}{l} 110110110110110110110110110110110110110110 \\ 111010011101001110100111010011101001110100 \\ \hline 110010010100000110100110010010100000110100 \end{array}$$

que tiene periodo 21, y podría ser obtenido con un LFSR de parámetros $D^6+D^4+D^2+D+1, 6$.

Función no lineal de secuencias

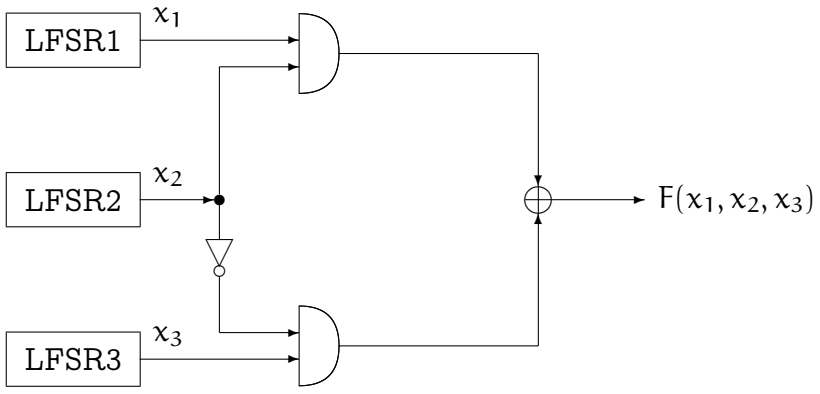
Supongamos que tenemos N secuencias, $x_1, x_2, \cdots, x_N$ obtenidas cada una a partir de un LFSR de periodo máximo, es decir, tenemos N sucesiones de bits, cada una obtenida con un polinomio primitivo $c_i(D)$, de grado L_i.

$$\begin{array}{ll} (c_1(D), L_1) & \longrightarrow \; x_{1\,1}\, x_{1\,2}\, x_{1\,3} \cdots \\ (c_2(D), L_2) & \longrightarrow \; x_{2\,1}\, x_{2\,2}\, x_{2\,3} \cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ (c_N(D), L_N) & \longrightarrow \; x_{N\,1}\, x_{N\,2}\, x_{N\,3} \cdots \end{array}$$

Si combinamos dichas secuencias mediante la función

$$F(x_1, x_2, \cdots, x_N) = a_0 + \sum_{i=1}^{N} a_i x_i + \sum_{i<j} a_{ij} x_i x_j + \cdots + a_{12\ldots N} x_1 x_2 \cdots x_N$$

Entonces obtenemos una secuencia cuya complejidad lineal es $F(L_1, L_2, \cdots L_N)$.
Como ejemplo, veamos el *Generador de Geffe*
Tenemos en este caso tres LFSR, de longitudes L_1, L_2 y L_3 y periodo $2^{L_1}-1$, $2^{L_2}-1$ y $2^{L_3}-1$.
Una representación gráfica de dicho generador sería



Es decir, la función de mezcla de los tres LFSR es $F(x_1, x_2, x_3) = x_1 \cdot x_2 + (1 + x_2)x_3 = x_1x_2 + x_2x_3 + x_3$, y por tanto, su complejidad lineal es $L_1L_2 + L_2L_3 + L_3$, y su periodo $mcm(2^{L_1} - 1, 2^{L_2} - 1, 2^{L_3} - 1)$.
Vamos por último a explicar brevemente el algoritmo A5, usado en el sistema global para las comunicaciones móviles (GSM)

Este algoritmo hace uso de tres LFSR, de parámetros $(x^{19}+x^{18}+x^{17}+x^{14}+1, 19)$, $(x^{22}+x^{21}+1, 22)$ y $(x^{23}+x^{22}+x^{21}+x^{8}+1, 23)$. Cada uno de los registros tiene un bit de reloj. En el primer registro es el noveno, en el segundo es el undécimo, mientras que en el tercero es el octavo. Llamemos a estos bits C_1, C_2 y C_3.
Los bits de reloj controlan que el registro se desplace o no. Si el bit de reloj de un registro coincide con la mayoría de los bits de reloj, contando los tres registros, entonces se produce un desplazamiento, mientras que si no coincide no se produce el desplazamiento. Esto garantiza que en cada pulso de reloj se desplacen 2 ó 3 registros (por ejemplo, si $C_1 = C_3 = 1$ y $C_2 = 0$, se desplazan los registros uno y tres, o si $C_1 = 1$ y $C_2 = C_3 = 0$, se desplazan los registros dos y tres). Esto podemos expresarlo mediante la función $F(C_1, C_2, C_3) = C_1C_2 + C_1C_3 + C_2C_3$. Si $C_i = F(C_1, C_2, C_3)$ entonces se desplaza el registro i-ésimo, mientras que si $C_i \neq F(C_1, C_2, C_3)$ entonces no se desplaza.

Parte III

Cifrado por bloques simétrico

CRIPTOGRAFÍA V

Generalidades

Se denominan así a una serie de algoritmos de cifrado con unas características comunes:

1. El mensaje a cifrar se divide en bloques de tamaño fijo (64 bits, 128 bits, etc.). Si la longitud del mensaje no es múltiplo del tamaño de bloque, se completa el último hasta alcanzar el tamaño necesario.

2. A cada uno de los bloques se le aplica el algoritmo de cifrado.

3. La clave de cifrado es la misma que la de descifrado (de ahí que se les llame simétricos). Esto obliga a que la clave tenga que mantenerse en secreto por el emisor y el receptor (de ahí que se les llame *criptosistemas de llave secreta o privada*).

La mayoría de estos sistemas se basan en diferentes capas de sustituciones-permutaciones aplicadas a cada uno de los bloques.

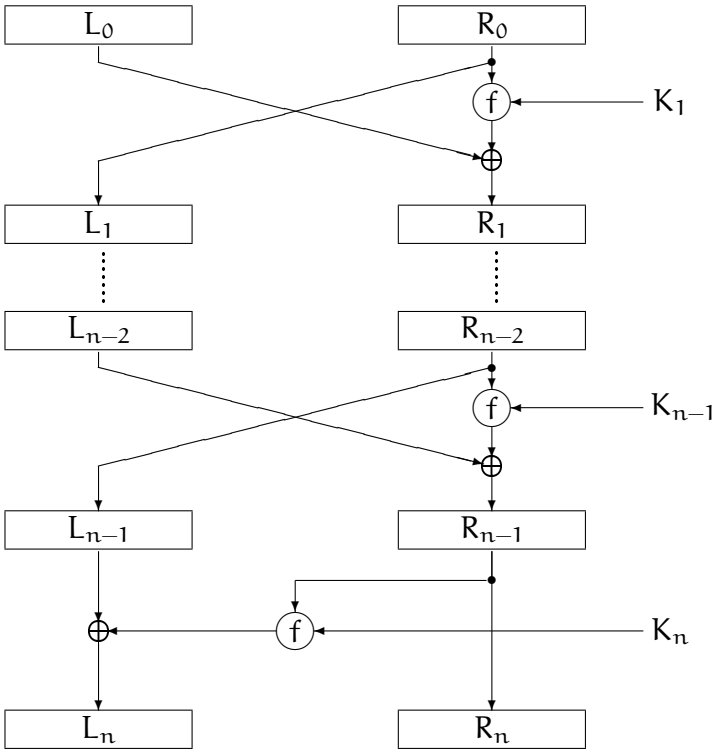
1 Redes de Feistel

Un método común para los cifradores en bloque fue diseñado por el criptógrafo alemán Horst Feistel a principios de los años 70. Feistel trabajaba para IBM, y fue el creador, junto con Don Copperschmit del criptosistema Lucifer, que ya incorporaba dicho sistema denominado *Red de Feistel*. El criptosistema Lucifer fue el precursor de uno de los criptosistemas simétricos más conocidos, estudiados y empleados: el DES, del que hablaremos próximamente.
Una red de Feistel consiste en dividir el bloque a cifrar en dos subbloques de igual tamaño, y se le aplican un número determinado de rondas, en las que la salida de una ronda sirve como entrada de la siguiente. En cada ronda se intercambian los bloques, y se suma el resultado de aplicar una función que depende del bloque precedente y de la clave (una clave para cada ronda). La última ronda es ligeramente diferente, al no haber intercambio de bloques.
Este método de cifrado tiene la propiedad de que para descifrar se emplea el mismo algoritmo que para cifrar, sólo que se utilizan las claves en orden inverso, independiente de la función que se emplee.
Vamos a precisar en que consiste una red de Feistel. Supongamos que n es el número de rondas. Tendremos entonces n claves $K_1, K_2, \cdots, K_n$

Partimos de un bloque M y lo dividimos en dos subbloques L_0 y R_0. Tras la primera ronda, obtenemos dos bloque L_1, R_1; tras la segunda L_2, R_2, y así sucesivamente hasta la penúltima en que obtenemos los bloques L_{n-1}, R_{n-1}. En todas estas rondas se emplea el mismo esquema. Tras la última ronda obtenemos los bloque L_n, R_n. Por último, se unen estos dos bloques.
El esquema que se utiliza para cada ronda lo podemos expresar como sigue:

$$\begin{array}{ll} L_i = R_{i-1} & R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \qquad \text{si } 1 \leq i \leq n-1 \\ L_n = L_{n-1} \oplus f(R_{n-1}, K_n) & R_n = R_{n-1} \end{array}$$

La siguiente figura nos da una representación gráfica de una red de Feistel.



Vamos a ver a continuación como el algoritmo de cifrado es igual al algoritmo de descifrado. Vamos a hacerlo, en primer lugar, en un caso de 3 rondas. Visto esto puede hacerse fácilmente en el caso general.
Supongamos que tenemos una red de Feistel con tres rondas. Necesitamos entonces 3 claves, K_1, K_2 y K_3. Partimos de dos bloque L_0 y R_0, y obtenemos los bloque L_1, R_1; L_2, R_2 y L_3, R_3 como sigue:

$$\begin{array}{ll} L_1 = R_0 & R_1 = L_0 \oplus f(R_0, K_1) \\ L_2 = R_1 & R_2 = L_1 \oplus f(R_1, K_2) \\ L_3 = L_2 \oplus f(R_2, K_3) & R_3 = R_2 \end{array}$$

La función de descifrado, se parte de dos bloque L'_0, R'_0 y aplicamos el mismo proceso, pero cambiando el orden de las claves.
Tenemos entonces:

$$\begin{array}{ll} L'_1 = R'_0 & R'_1 = L'_0 \oplus f(R'_0, K_3) \\ L'_2 = R'_1 & R'_2 = L'_1 \oplus f(R'_1, K_2) \\ L'_3 = L'_2 \oplus f(R'_2, K_1) & R'_3 = R'_2 \end{array}$$

Veamos que si $L'_0 = L_3$ y $R'_0 = R_3$ entonces $L'_3 = L_0$ y $R'_3 = R_0$.

$L'_1 = R'_0 = R_3 = R_2$;
$R'_1 = L'_0 \oplus f(R'_0, K_3) = L_3 \oplus f(R_3, K_3) = (L_2 \oplus f(R_2, K_3)) \oplus f(R_2, K_3) = L_2$;

$L'_2 = R'_1 = L_2 = R_1$;
$R'_2 = L'_1 \oplus f(R'_1, K_2) = R_2 \oplus f(L_2, K_2) = (L_1 \oplus f(R_1, K_2)) \oplus f(R_1, K_2) = L_1$;

$L'_3 = L'_2 \oplus f(R'_2, K_1) = R_1 \oplus f(L_1, K_1) = (L_0 \oplus f(R_0, K_1)) \oplus f(R_0, K_1) = L_0$;
$R'_3 = R'_2 = L_1 = R_0$;

En general, se tiene que si $(L_0, R_0),\ (L_1, R_1), \cdots (L_i, R_i), \cdots (L_{n-1}, R_{n-1}),\ (L_n, R_n)$ son los distintos bloques que van apareciendo en las distintas etapas de una red de Feistel con valores iniciales (L_0, R_0); $(L'_0, R'_0),\ (L'_1, R'_1), \cdots (L'_i, R'_i), \cdots (L'_{n-1}, R'_{n-1}),\ (L'_n, R'_n)$ son los bloques que aparecen al aplicar la misma red de Feistel, con valores iniciales (L'_0, R'_0) pero cambiando el orden de las claves; y $L'_0 = L_n$ y $R'_0 = R_n$, entonces:

$$\begin{array}{lll} L'_i = R_{n-i} & R'_i = L_{n-i} & \text{si } 1 \leq i \leq n-1 \\ L'_n = L_0 & R'_n = R_0. & \end{array}$$

Veamos el porqué.

$L'_1 = R'_0 = R_n = R_{n-1}$;
$R'_1 = L'_0 \oplus f(R'_0, K_n) = L_n \oplus f(R_n, K_n) = (L_{n-1} \oplus f(R_{n-1}, K_n)) \oplus f(R_{n-1}, K_n) = L_{n-1}$.

Para i comprendido entre 2 y $n-1$ se tiene que:

$L'_i = R'_{i-1} = L_{n-(i-1)} = L_{n-i+1} = R_{n-i}$;
$R'_i = L'_{i-1} \oplus f(R'_{i-1}, K_{n-i+1}) = R_{n-i+1} \oplus f(L_{n-i+1}, K_{n-i+1}) = (L_{n-i} \oplus f(R_{n-i}, K_{n-i+1})) \oplus f(R_{n-i}, K_{n-i+1}) = L_{n-i}$.

Y por último,

$L'_n = L'_{n-1} \oplus f(R'_{n-1}, K_1) = R_1 \oplus f(L_1, K_1) = (L_0 \oplus f(R_0, K_1)) \oplus f(R_0, K_1) = L_0$;
$R'_n = R'_{n-1} = L_1 = R_0$.

Algunos de los sistemas de cifrado que emplean redes de Feistel son Lucifer, FEAL, CAST y sobre todo el DES, que ya tiene un capítulo importante en la historia de la Criptografía.
Antes de estudiar en detalle el DES, vamos a describir diversos modos de cifrado en bloque.

2

Modos de operación para algoritmos de cifrado por bloques

Vamos a ver en esta sección varios modos en que se puede operar para cifrados por bloques. Hay cuatro que son los más conocidos y usados, aunque se han propuesto algunos más.

La diferencia entre estos modos de operación está en el tratamiento que se da a los bloques. Supongamos que tenemos un mensaje a cifrar, y que este mensaje está dividido en bloques. Vamos a denotar estos bloques como $M(1), M(2), \cdots, M(n), \cdots$. Denotaremos por $C(1), C(2), \cdots, C(n), \cdots$ los distintos bloques de criptograma que se obtienen en el proceso de cifrado.
Vamos a llamar $E(x, y)$ al resultado de aplicar el algoritmo de cifrado que estamos usando al bloque de información x, con la clave y, mientras que llamaremos $D(x, y)$ al resultado de aplicar el algoritmo de descifrado al bloque de criptograma x con la clave y.
Los cuatro modos básicos de cifrado son:

1. ECB (Electronic Code Book).
2. CBC (Cipher Block Chaining).
3. CFB (Cipher Feed Back).
4. OFB (Output Feed Back).

. . . . . 2.1

Electronic Code Book

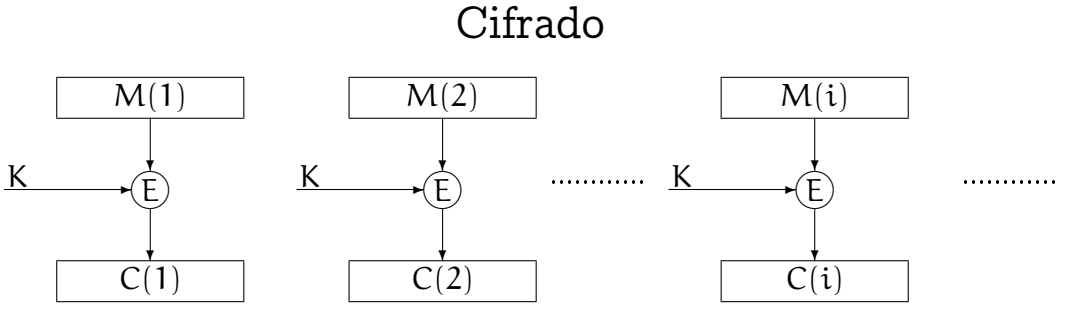
Es el modo más sencillo de aplicar un algoritmo de cifrado por bloques. El algoritmo de cifrado se aplica de manera independiente a cada uno de los bloques del mensaje. Es decir, si K es la clave de cifrado se tiene:

$$C(i) = E(M(i), K) \quad \text{para } i = 1, 2, \cdots$$

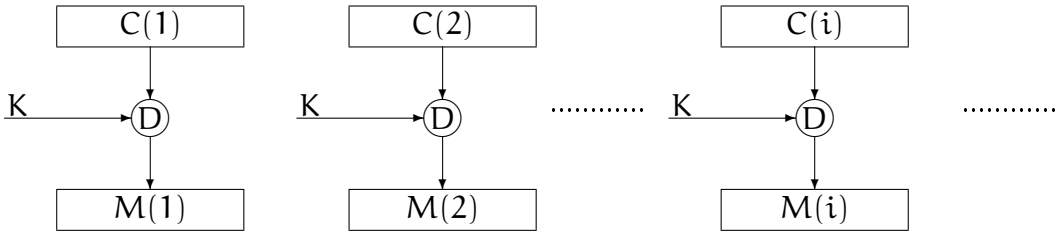
Para descifrar se opera de la misma forma, pero usando la función de descifrado:

$$M(i) = D(C(i), K) \quad \text{para } i = 1, 2, \cdots$$

Gráficamente, responde al siguiente esquema:



Descifrado



Entre las ventajas de este modo está que se pueden cifrar bloques independientemente de su orden, lo cual es adecuado para bases de datos o ficheros en los que se requiera un acceso aleatorio. También es resistente a errores, pues un error afecta únicamente al bloque en que se ha producido, y no afecta al resto del mensaje.

Sin embargo, presenta un grave inconveniente, sobre todo si el mensaje presenta patrones repetidos (por ejemplo, para imágenes, o correo electrónico), pues dichos patrones se pueden mantener en el mensaje cifrado.

También presenta el riesgo de la sustitución por bloques. Un atacante puede cambiar bloque y alterar los mensajes, incluso sin conocer la clave empleada.

. . . . . 2.2

Cipher Block Chaining

En este caso, el algoritmo de cifrado no se aplica independiente a cada uno de los bloques, sino que el resultado de cifrar un bloque se usa como dato de entrada para cifrar el bloque siguiente. Lo que se hace es que se le suma al bloque de mensaje que va a continuación.

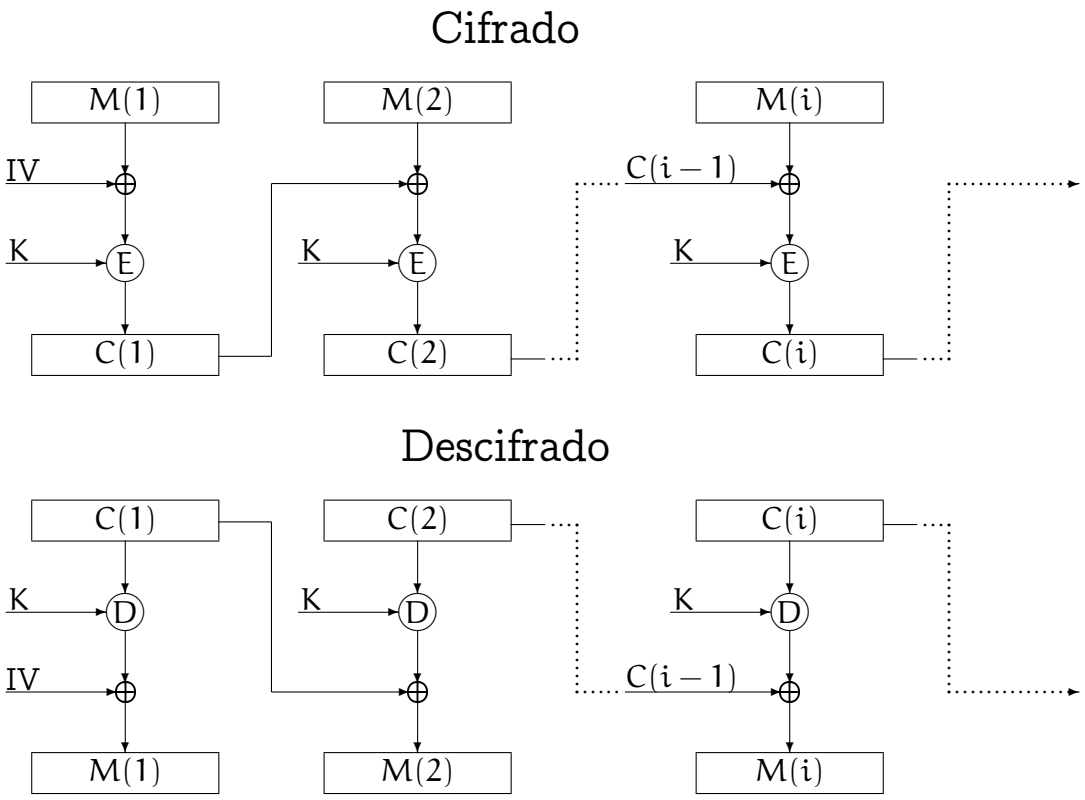
Para iniciar, al primer bloque de mensaje se le suma un bloque del mismo tamaño, aleatorio, que se denomina IV (Inicianilation Vector). La dinámica de este modo de cifrado viene dada por:

$$\begin{aligned} C(0) &= IV \\ C(i) &= E(M(i) \oplus C(i-1), K) \qquad \text{para } i = 1, 2, \cdots \end{aligned}$$

mientras que el descifrado sería:

$$M(i) = C(i-1) \oplus D(C(i), K) \quad \text{para } 1 = 1, 2, \cdots$$

Esquemáticamente, lo podemos representar:



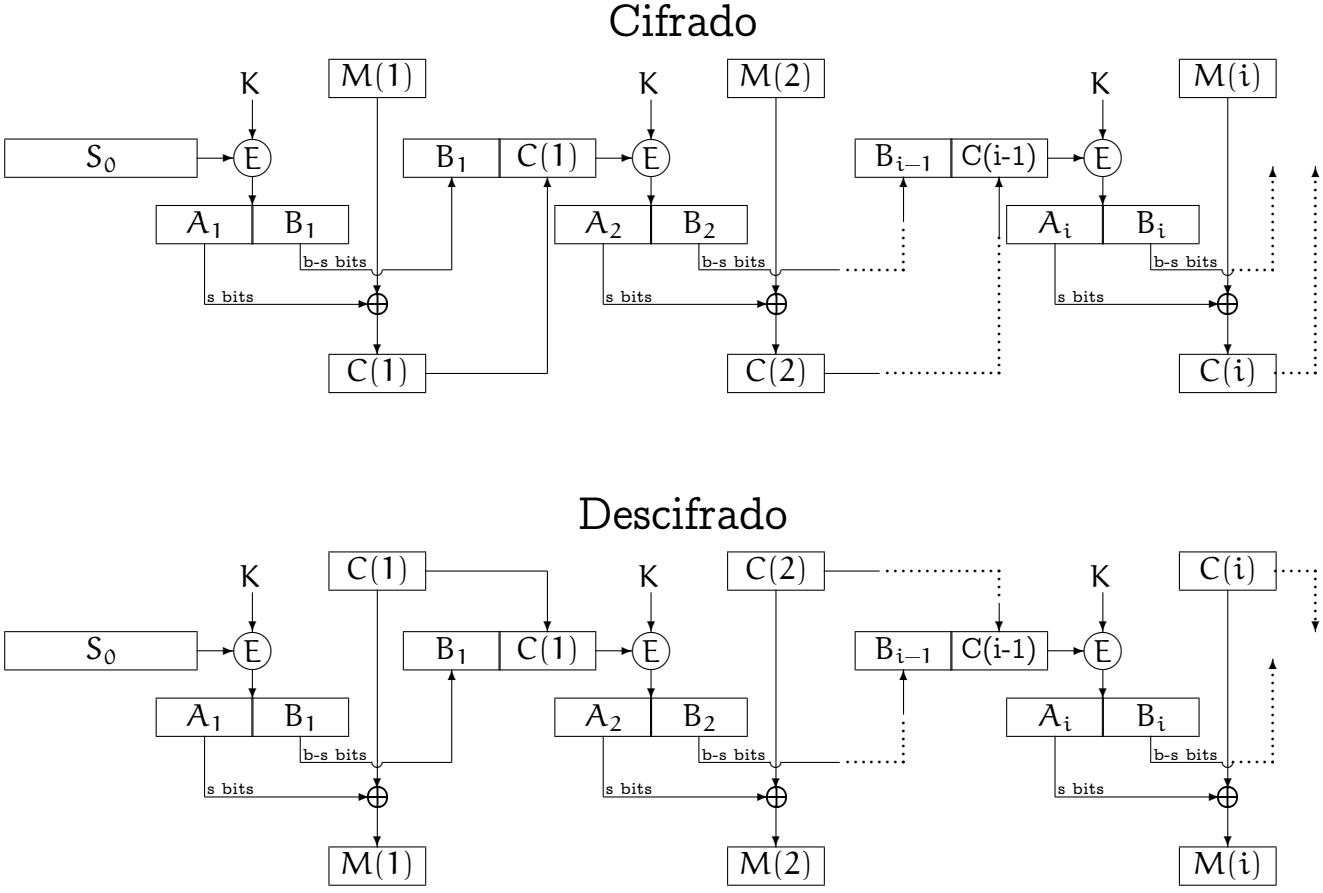
El mismo mensaje puede tener cifrados diferentes. Basta con elegir un vector de inicialización distinto en cada caso. Además, ahora si se sustituye un bloque de criptograma por otro, o se insertan algunos bloques, eso afectaría al resto de los bloques, lo que imposibilita realizar dicha acción sin que sea detectado el cambio.

. . . . . 2.3

Cipher Feed Back

En este caso, además del tamaño de los bloque con los que opera el algoritmo de cifrado/descifrado, y que llamaremos b, necesitamos otro parámetro, que será un número entero s tal que $1 \leq s \leq b$. Entonces, dividimos el mensaje en bloques de tamaño s, y el criptograma nos saldrá también en bloques de tamaño s.
Dividimos el mensaje en bloque de tamaño s. Partimos entonces de un bloque inicial formado por b bits, que denominaremos S_0, y utilizaremos dos funciones R_x y L_x, donde x es un entero, y aplicadas a un bloque de tamaño b nos devuelve, la primera, un bloque formado por los x bits situados más a la derecha (los últimos), o los x bits situados más a la izquierda. A veces se suele llamar a estas funciones LSB_x y MSB_x respectivamente, pues hacen referencia a los bits menos significativos y los bits más significativos.
A partir de estos datos, obtenemos $C(1)$ como $M(1) \oplus L_s(E(S_0, K))$, y obtenemos un nuevo bloque S_1 como la concatenación de los bloques $R_{b-s}(E(S_0, K))$ y $C(1)$.
Una vez obtenido S_1, y con $M(2)$ calculamos $C(2)$ y S_2 de la misma forma. En resumen podemos escribir que:
$C(i) = M(i) \oplus L_s(E(S_{i-1}, K))$;

$S_i = (R_{b-s}(E(S_{i-1}, K)), C(i))$.
Para descifrar, partimos de S_0, y $C(1), C(2), \cdots$ y tenemos:
$M(i) = C(i) \oplus L_s(E(S_{i-1}, K))$;
$S_i := (R_{b-s}(E(S_{i-1}, K), C(i)))$.

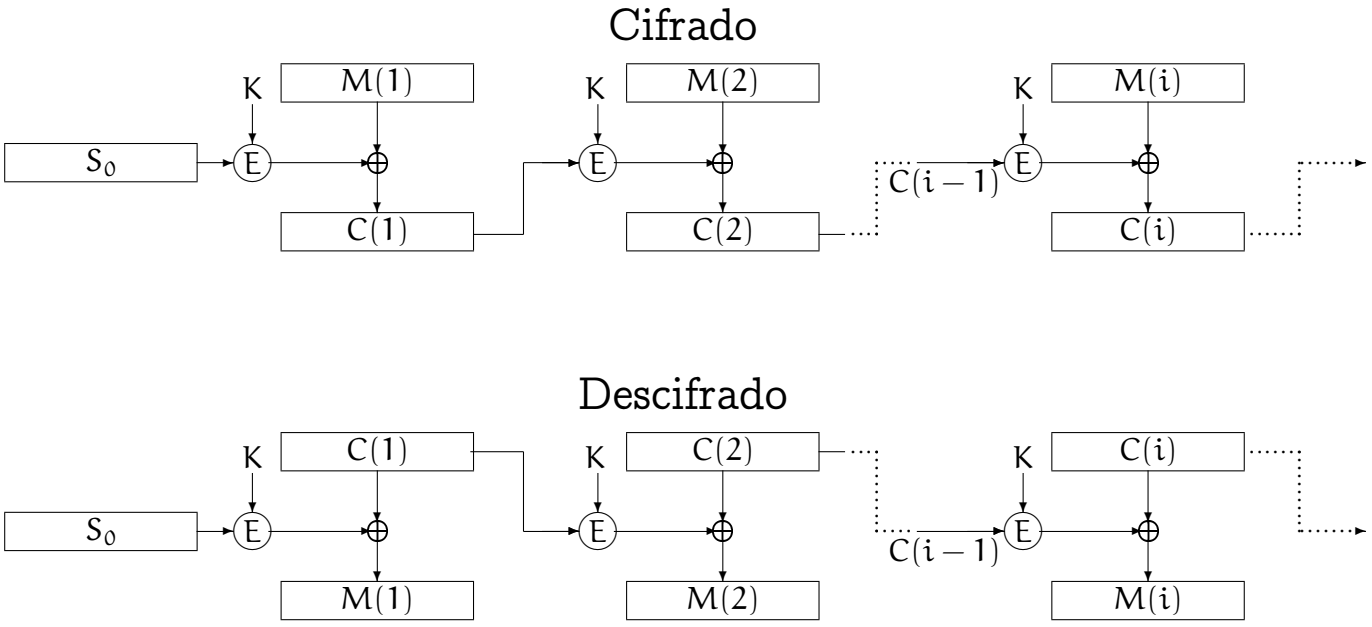


El caso más sencillo es cuando $s = b$. Es inmediato que $L_b = Id$, mientras que R_0 nos devuelve el bloque vacío. Además, como los bloques en que se divide el mensaje son los más grande posible, el proceso de cifrado y descifrado es más rápido.
En este caso, también tendríamos un bloque inicial S_0, y el proceso de cifrado vendría dado por:

$$C(0) = S_0$$
$$C(i) = M(i) \oplus E(C(i-1), K) \qquad \text{para } i = 1, 2, \cdots$$

mientras que el descifrado sería:

$$M(i) = C(i) \oplus E(C(i), K) \qquad \text{para } 1 = 1, 2, \cdots$$



. . . . . 2.4

Output Feed Back

En este modo, lo que se hace es sumarle a cada bloque del mensaje un bloque del mismo tamaño, obtenido cifrando sucesivamente un bloque inicial mediante el algoritmo de cifrado. Este modo podría ser como un cifrado en flujo, en el que la secuencia pseudoaleatoria es obtenida mediante aplicación sucesiva de un algoritmo, a partir de un vector o bloque inicial que podría ser considerado como la semilla.

Partimos pues de un vector de inicialización IV, y a partir de él construimos la secuencia $R(0), R(1), \cdots$ como sigue:

$$R(0) = IV \qquad R(i) = E(R(i-1), K) \text{ para } i = 1, 2, \cdots$$

Y con esta secuencia, ciframos.

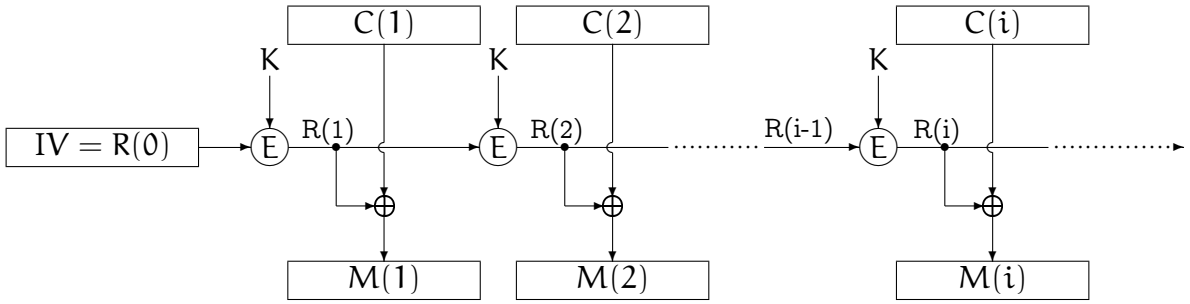
$$C(i) = M(i) \oplus R(i)$$

Nótese que la secuencia $R(i)$ puede ir calculándose conforme se va cifrando, luego no es necesario almacenarla entera.

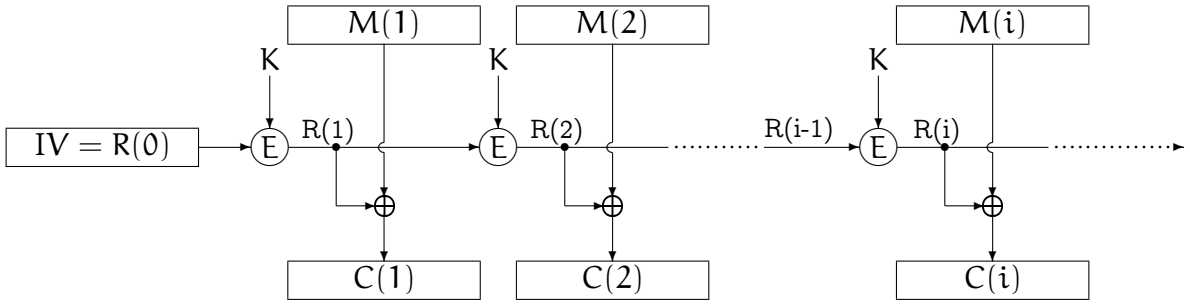
Para descifrar, tenemos el vector inicial IV, y los criptogramas $C(1), C(2), \cdots$. Entonces podemos obtener el mensaje en claro de la siguiente forma.

$$R(0) = IV \qquad R(i) = E(R(i-1), K) \text{ para } i = 1, 2, \cdots$$
$$M(i) = C(i) \oplus R(i)$$

Cifrado



Descifrado



Si un atacante conociera un bloque de texto en claro $M(i)$ y su correspondiente cifrado $C(i)$, entonces podría reemplazar el bloque correspondiente $M(i)$ por otro bloque $M'(i)$. Para esto, únicamente tendría que sustituir el criptograma $C(i)$ por $C'_i = C(i) \oplus M(i) \oplus M'(i)$.

CRIPTOGRAFÍA VI

Data Encryption Standard (DES)

A comienzos de los años 70 se empezó a producir una proliferación de los ordenadores. A su vez, se desarrollaron nuevos sistemas de comunicación que permitieron a bancos y empresas almacenar y transmitir un volumen de información cada vez mayor. Estas organizaciones demandaron entonces una protección de la información. Entonces, la NBS (National Bureau of Standars) puso en marcha un programa con el objetivo de dotar de seguridad a las comunicaciones y bases de datos. Dentro de este programa se incluía la adopción de un estándar de cifrado de uso general. Así, en mayo de 1973, la NBS solicitó propuestas para tal estándar, mediante convocatoria pública en el Registro Federal de los Estados Unidos.
Los requisitos que debía satisfacer este estándar son:

- El algoritmo debería proporcionar un elevado grado de seguridad.
- El algoritmo debe estar completamente especificado y ser fácil de entender.
- La seguridad del algoritmo dependerá exclusivamente de la llave y no del secreto del mismo.
- El algoritmo deberá estar disponible para todos los usuarios.
- El algoritmo deberá ser adaptable para su utilización en diversas aplicaciones.
- El algoritmo deberá ser fácilmente implementado en sistemas electrónicos.
- El algoritmo deberá ser eficiente.
- El algoritmo deberá se exportable.

En principio se recibieron muy pocas propuestas, y ninguna fue aceptada por la NBS, así que en agosto de 1974 la NBS efectuó un segundo concurso. En esta ocasión la NBS recibió una propuesta de IBM, basada en el algoritmo Lucifer, y que consideró interesante. Antes de adoptarlo como estándar lo sometió a la consideración de la Agencia Nacional de Seguridad (NSA).
Tras una evaluación positiva de los expertos de la NSA, en marzo de 1975 la NBS publicó en el Registro Federal los detalles del método de cifrado presentado por IBM para someterlo a discusión pública. Dos años después, en enero de 1977, fue adoptado por la NBS como

estándar de cifrado con el nombre de *Data Encryption Standard* (DES). Desde entonces ha estado sometido a multitud de estudios. La NBS (actualmente NIST - National Institute of Standars and Technology -) revisa el método cada 5 años.
Desde poco después de su publicación, compañías como Motorola, AMD o DEC iniciaban la producción de y distribución de chips que implementaban el DES.
Pero la propuesta de DES no ha estado exenta de polémica en los Estados Unidos. El motivo es la intervención de la NSA en el proceso. Aún hoy no se conoce con exactitud si la NSA modificó el proyecto presentado por IBM. Algunos técnicos de IBM afirman que la NSA no alteró lo más mínimo el algoritmo, pero otros afirman lo contrario. Sí hubo, sin embargo, dos modificaciones con respecto al predecesor, el cifrado Lucifer.
La primera modificación es la reducción del tamaño de los bloques y de la clave, que pasó de 128 bits a 64 en el caso de los bloques de cifrado, y a 56 en el caso de la clave. Esta disminución en el tamaño de la clave ha levantado sospechas. Una clave de 56 bits daba seguridad suficiente para los ordenadores de entonces. Pero quizá fueran accesibles para los supercomputadores de la NSA, que de esta forma tendría acceso a múltitud de mensajes privados.
La segunda modificación se encuentra en las S-cajas. En ellas se basa toda la seguridad del DES. En cada una de las cajas entran 6 bits, que se transforman en 4 de forma misteriosa. Mucha gente sospechó que en ellas se escondía una "puerta trasera" que sólo la Agencia sabría como abrir. La NSA siempre lo negó. Lo que es cierto es que las cajas proporcionan al cifrado DES una seguridad mucho mayor que la que tenía Lucifer, a pesar de tener una clave mucho menor.
El número "reducido" de claves (2^{56}) invitaba a un ataque por fuerza bruta. De hecho, ha sido el único método efectivo para derrotar a DES. Se han descubierto otros métodos con menos coste computacional, pero ninguno ha sido viable en la práctica. Los más conocidos son los denominados *criptoanálisis diferencial* y *criptoanálisis lineal*.
El primero fue dado a conocer por los israelíes Eli Bihman y Adi Shamir en 1990. Este método se ha mostrado eficiente contra el Lucifer, o contra una versión de DES con no más de 8 rondas, pero no contra la versión auténtica. De hecho, después de publicar Biham y Shamir su trabajo, el miembro del equipo IBM que diseñó el DES Don Coppersmith afirmó que ellos conocían la técnica del criptoanálisis diferencial, y por esa razón le pusieron 16 rondas al DES.
El segundo fue ideado por Mitsuru Matsui en 1993. Contra el DES es algo más eficaz que el criptoanálisis diferencial, pero en la práctica requiere una gran cantidad de texto. En esta ocasión, Coppersmith dijo no conocer el método de Matsui cuando diseñaron el DES.
Una de las ventajas de DES, por su sencillez y simplicidad, es que utiliza los criptosistemas clásicos.
La combinación de varios criptosistemas debe ir encaminada a hacer desaparecer cualquier información redundante que proporcione el texto cifrado, por ejemplo el estudio de frecuencias; o el reconocimiento de patrones dentro del lenguaje.
Algunas de las características del DES podemos resumirlas en:

1. Cifra bloques de 64 bits de texto y no altera el tamaño del bloque en el proceso de su

cifrado.

2. El algoritmo de cifrado coincide con el de descifrado, salvo que en ambos procesos las listas de las llaves son distintas (*para el descifrado se utiliza la misma serie de claves pero en orden inverso*).

3. Cada llave es una lista de 56 bits, aunque se da como una lista de 64 bits. Lo que ocurre en realidad es que los bits que usan las posiciones: 8, 16, 24, 32, 40, 48, 56 y 64 son utilizadas para el control de paridad y, salvo esto, ignoradas en el proceso.

4. No hay restricciones para las llaves aparte de su tamaño y pueden ser cambiadas en cualquier momento.

5. Hay llaves que se consideran débiles, pero pueden ser fácilmente detectadas y por tanto se pueden despreciar.

6. Repite 16 veces un mismo proceso al que denominaremos *ronda* y las operaciones que emplean no van más allá de la aritmética estándar, operaciones del Álgebra de Boole y permutaciones. En los años 70 se implementó en hardware mientras que las implementaciones en software, que ahora han mejorado, eran bastante toscas. El esquema repetitivo del algoritmo lo hacen muy adecuado para usarlo en un chip de propósito específico.

Vamos a describir a continuación el DES. Para ello, vamos a dividir el algoritmo en tres procesos que explicaremos en detalle:

- El proceso de cifrado

- La función f. Como sabemos el DES utiliza redes de Feistel, en las que interviene una función f que sa va sumando a algunos de los subbloques que nos van apareciendo.

- La expansión de claves. A partir de la clave inicial, hemos de obtener una clave para cada una de las ronda.

1 Cifrado con DES

El DES opera con bloques de 64 bits. Dado un bloque de texto plano de 64 bits, el algoritmo DES nos devuelve un bloque cifrado el mismo tamaño, y que depende tanto del bloque inicial como de la clave.

Se comienza con una permutación (reordenación) de los 64 bits del mensaje. Llamaremos a esta permutación PI (Permutación Inicial) La permutación viene por la siguiente tabla:

Permutación Inicial (PI)							
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

La tabla indica que el bit que ocupa la posición 58 del mensaje pasa a ser el primero después de la permutación. El que ocupa la posición 50 del mensaje pasa a ser el segundo tras la permutación, y así sucesivamente, leyendo de izquierda a derecha y de arriba a abajo.
Una vez transformado el bloque inicial con la permutación, dividimos el bloque en dos subbloque de 32 bits cada uno, L_0 y R_0 y seguimos el proceso descrito en la sección dedicada a redes de Feistel. Realizamos 16 rondas, y obtenemos al final dos subbloques L_{16}, R_{16}, que los juntamos y le aplicamos la inversa de la permutación inicial, que viene dada por la siguiente tabla:

Permutación Inicial Inversa (PI^{-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

De esta forma, el algoritmo de cifrado y descifrado es el mismo (salvo en el orden de las claves), pues para descifrar partiríamos del criptograma, y habría que aplicarle la inversa de la Permutación Inicial Inversa, es decir, la Permutación Inicial. Después habría que invertir la red de Feistel, que ya vimos que era únicamente invertir el orden de las claves, y por último la inversa de la Permutación Inicial.

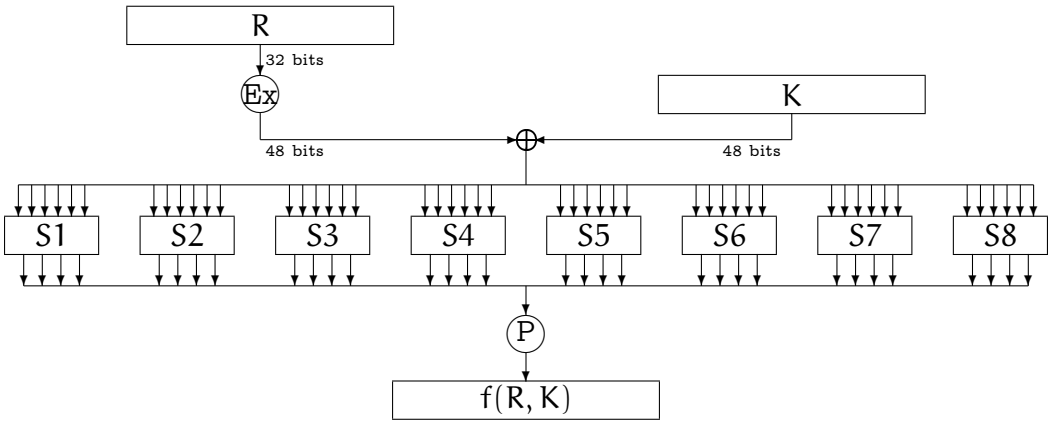
. . . . . 1.1

La función f

La función f tiene como entradas los 32 bits del bloque R_i, y los 48 bits de la clave de ronda K_{i-1}
Supongamos que tenemos R un bloque de 32 bits y K un bloque de 48 bits. Entonces hemos de hacer lo siguiente:

1. Del bloque de 32 bits obtenemos un bloque de 48 bits, mediante una Función de Expansión.
2. A los 48 bits obtenidos le hacemos una suma bit a bit con los 48 bits del bloque K. Tenemos entonces un bloque de 48 bits.
3. Dividimos el bloque en 8 subbloques de 6 bits cada uno.
4. Los seis bits de cada subloque son los datos de entrada para una S-caja, que devuelve como entrada 4 bits.
5. Se unen todos los bits que se han obtenido como salida de las S-cajas, resultando un bloque de 32 bits.
6. Se le aplica una permutación a los 32 bits.

El resultado de aplicar la última permutación es $f(R, K)$.
Veamos esquemáticamente el funcionamiento de f.



Vamos a detallar en qué consisten las distintas funciones que nos han aparecido.
Por una parte, tenemos la función de expansión. Ésta tiene como entrada 32 bits y como salida un bloque de 48 bits. Podemos representarla en la siguiente tabla:

Función de Expansión (Ex)					
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

con el mismo significado que para la Permutación Inicial.
La permutación P que aparece al final de la función f viene dada por:

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

y por último, las 8 S-cajas:

$\mathbf{S_1}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	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

$\mathbf{S_2}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	5
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

$\mathbf{S_3}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	5	4	3	11	5	2	12

$\mathbf{S_4}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

$\mathbf{S_5}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	2	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	7	13	6	15	0	9	10	4	5	3

S_6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

El mecanismo de las S-cajas es el siguiente:
A cada S-caja entran 6 bits b_1 b_2 b_3 b_4 b_5 b_6. Consideramos los números binarios b_1b_6 y $b_2b_3b_4b_5$, que en decimal representan un número entre 0 y 3, en el primer caso, y un número entre 0 y 15 en el segundo.
Buscamos en la S-caja correspondiente la fila que tiene al inicio el número b_1b_6 y la columna que tiene al principio el número $b_2b_3b_4b_5$, y en la intersección de ambas, tenemos un número entre 0 y 15, que una vez pasado a binario nos da los cuatro bits de salida de la S-caja.
La robustez del DES está en las S-cajas. Los cambios en ellas no hacen otra cosa que debilitarlo. En su diseño se han tenido en cuenta diferentes aspectos:

1. Las filas de cada caja son una permutación de los elementos del conjunto

$$\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$$

.

2. La salida de ninguna caja es una función lineal o afín de la entrada.

   Por ejemplo, si fuera una función lineal se tendría que $S(x \oplus y) = S(x) \oplus S(y)$. Vamos a tomar la caja 1, $x = 010110$ e $y = 110101$. Puesto que $00)_2 = 0$ y $1011)_2 = 11$, para calcular $S_1(x)$ tenemos que buscar la fila 0 y la columna 11, y nos da $12 = 1100)_2$. Por tanto, $S(x) = 1100$.

   Para calcular $S_1(y)$ tenemos en cuenta que $11)_2 = 3$ y $1010)_2 = 10$, luego $S_1(y) = 0011$.

   Ahora bien $x \oplus y = 100011$, luego $S_1(x \oplus y) = 1100$, mientras que $S_1(x) \oplus S_1(y) = 1100 \oplus 0011 = 1111$.

   Para que S_1 sea una función afín se debe verificar que la función g dada por $g(x) = S_1(x) - S_1(0)$ sea lineal. Al estar en $\mathbb{Z}_2$ podemos poner $g(x) = S_1(x) \oplus S_1(0) = S_1(x) \oplus 1110$.

Tomamos $x = 101101$ e $y = 011011$. Para calcular $S_1(x)$ hemos de mirar en la fila 3 y la columna 6, mientras que para calcular $S_1(y)$ hay que mirar en la fila 1 y la columna 13. Por tanto, se tiene que $S_1(x) = 0001$ y $S_1(y) = 0101$.

Puesto que $x \oplus y = 110110$, entonces hemos de buscar en la fila 2 y la columna 11, lo que nos da $S_1(x \oplus y) = 0111$. Por tanto:

$$g(x) \oplus g(y) = S_1(x) \oplus 1110 \oplus S_1(y) \oplus 1110 = 0001 \oplus 0101 = 0100;$$
$$g(x \oplus y) = S_1(x \oplus y) \oplus S_1(0) = 0111 \oplus 1110 = 1001.$$

Y como podemos ver son distintos.

Hemos visto como la salida de la primera caja no es lineal ni afín. Lo mismo ocurre con el resto.

3. $S(x)$ y $S(x \oplus 001100)$ deben diferir en al menos dos bits.

   Por ejemplo, tomamos $x = 100100$. Entonces $x \oplus 001100 = 101000$, luego $S_2(x) = 0111$ y $S_2(x \oplus 001100) = 1010$, que se diferencian en tres bits.

   En el caso $x = 000010$, se tiene que $S_2(x)$ y $S_2(x \oplus 001100)$ se diferencian en dos bits ($S_2(x) = 0001$ y $S_2(x \oplus 001100) = 0100$).

   También se pueden diferenciar en los cuatro bits, como es el caso de $S_2(100010)$ y $S_2(100010 \oplus 001100)$.

4. $S(x) \neq S(x \oplus 11uv00)$ para cualesquiera u, v.

   Por ejemplo, sea $x = 101110$. Entonces $x \oplus 11uv00$ puede tomar los valores 010010, 010110, 011010 y 011110. Si calculamos sus transformados por S_3 obtenemos:

$$S_3(101110) = 0000$$
$$S_3(010010) = 1101 \quad S_3(010110) = 0111 \quad S_3(011010) = 0100 \quad S_3(011110) = 1000$$

5. Si un bit de entrada se mantiene constante, entonces se minimiza la diferencia entre el número de ceros y el número de unos a la salida.

6. Si se fija uno de los bits de entrada a una caja, y se observa un bit de posición fija en la salida, entonces el número de entradas pra las que el bit de salida es 0 es aproximadamente igual al número de entradas para las que el bit de salida es 1.

   Por ejemplo, nos fijamos la caja sexta, y fijamos el quinto bit a 1, es decir, consideramos las 32 entradas de la forma $----1-$. Las 32 salidas son:

0001 1111 0010 1000 1101 0100 0111 1011 1111 0010 1100 0101 0001 1110 1011 1000
1110 0101 1000 0011 0000 1010 1101 0110 0011 1100 0101 1010 1110 0111 0000 1101

Si nos fijamos en el cuarto bit de salida, vemos que hay 16 que valen 0, y 16 que valen 1. Lo mismo ocurre si nos fijamos en el tercero. Sin embargo, si nos fijamos en cualquiera de los dos primeros veremos que hay 15 ceros y 17 unos.

Si, por ejemplo, hubiéramos fijado el sexto bit, obtendríamos en los cuatro casos 16 bits iguales a cero, y 16 bits iguales a uno.

. . . . . 1.2

Generación de subclaves

La clave inicial del DES tiene 64 bits, de los cuales los que ocupan una posición mútiplo de 8 son ignorados. Por tanto, la clave es de 56 bits.
A partir de la clave se han de generar 16 claves, una para cada ronda, y que será un argumento de entrada para la función f. Cada una de estas subclaves tendrá 48 bits. Para generarla, se parte de la clave inicial de 64 bits y se le aplica una permutación PC1, que viene detallada en la siguiente tabla:

PC1						
57	49	41	33	56	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

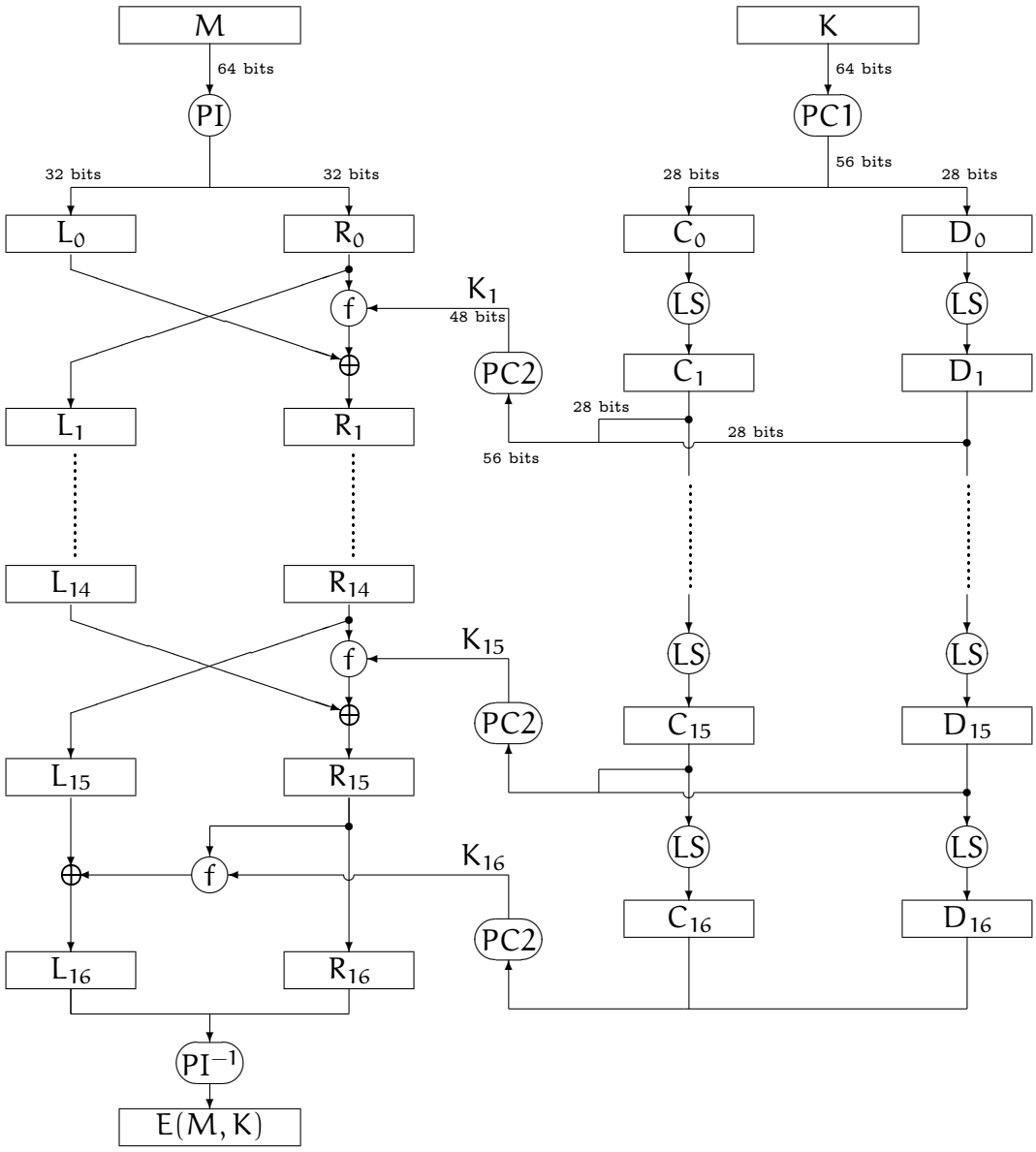
Obsérvese como los bits que ocupan las posiciones 8, 16, 24, 32, 40, 48, 56, 64 no se han tenido en cuenta.
Con los 56 bits se forman dos bloque C_0 y D_0, que tienen 28 bits cada uno.
Suponiendo que tenemos C_{i-1} y D_{i-1}, $1 \leq i \leq 16$, obtenemos los bloque C_i y D_i como $C_i = LS(C_{i-1})$ y $D_i = LS(D_{i-1})$, donde LS es un desplazamiento circular a la izquierda de uno o dos bits, dependiendo de la iteración en que nos encotremos. La siguiente tabla nos muestra, en función del la iteración, cuantas posiciones hay que deplazar a la izquierda los bits de los bloque C_i y D_i:

Iteración n°	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N° de bits desplazados	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	1

Los dos bloque se unen, resultando un bloque de 56 bits, y se le aplica la permutación PC2, que también selecciona 48 de los 56 bits. El resultado es la clave K_i, es decir, $K_i = PC2(C_i, D_i)$.
La permutación PC2 viene descrita en la siguiente tabla:

PC2					
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

En resumen, el proceso de cifrado de DES lo podemos visualizar como sigue:



1.3 *Llaves débiles y criptoanálisis*

Dentro de las 2^{56} claves que podemos utilizar en el DES, existen algunas ante las cuales el texto cifrado puede ser vulnerable. Son claves para las que la función de cifrado y descifrado son las mismas, es decir, se verifica, para cualquier mensaje M que $E_K(E_K(M)) = M$.
Sabemos que el descifrado con DES se lleva a cabo con el mismo algoritmo, pero con las claves tomadas en sentido inverso. Se conocen cuatro claves débiles de DES, que son las que, una vez aplicada transformación PC1, y dividido el resultado en dos bloques, estos dos bloque están compuestos por, bien todo ceros, bien todo unos. Estas cuatro claves son (en hexadecimal)

01 01 01 01 01 01 01 01	FE FE FE FE FE FE FE FE
E0 E0 E0 E0 F1 F1 F1 F1	1F 1F 1F 1F 0E 0E 0E 0E

En estas, a la hora de cifrar, se genera la misma clave para cada ronda.
Existen otras claves, llamadas semidébiles, que van por parejas, y que verifican que $E_{K_1}(E_{K_2}(M)) = M$, es decir, la función de descifrado con una clave coincide con la función de cifrado con la otra.
En estas, se generan sólo dos claves de ronda diferentes. Se conocen seis pares de este tipo de claves, que son:

(01 FE 01 FE 01 FE 01 FE, FE 01 FE 01 FE 01 FE 01)
(1F E0 1F E0 1F E0 1F E0, E0 1F E0 1F E0 1F E0 1F)
(01 E0 01 E0 01 F1 01 F1, E0 01 E0 01 F1 01 F1 01)
(1F EF 1F EF 0E FE 0E FE, FE 1F FE 1F FE 0E FE 0E)
(01 1F 01 1F 01 0E 01 0E, 1F 01 1F 01 0E 01 0E 01)
(E0 FE E0 FE F1 FE F1 FE, FE E0 FE E0 FE F1 FE F1)

El DES sabemos que es un algoritmo de cifrado que cifra bloque de 64 bits, y devuelve bloques del mismo tamaño. Vamos a denotar por $\mathcal{M}$ al conjunto de los posibles mensajes ($\mathcal{M}$ tiene 2^{64} elementos). Entonces, para cada clave K, tenemos una biyección

$$E_K : \mathcal{M} \to \mathcal{M}$$

Vamos a denotar por $S_{2^{64}}$ el conjunto de todas las biyecciones de $\mathcal{M}$ en $\mathcal{M}$. Entonces, para cada clave K se tiene que $E_K \in S_{2^{64}}$. Si llamamos $\mathcal{K}$ al conjunto de claves (este conjunto tiene 2^{56} elementos), lo que tenemos ahora es una aplicación inyectiva (pues dos claves diferentes no cifran igual todos los mensajes)

$$\mathcal{K} \to S_{2^{64}}$$

Tenemos entonces un subconjunto, de 2^{56} elementos, del grupo $S_{2^{64}}$, y la pregunta es si este conjunto tiene estructura de grupo. Dicho de otra forma, si dadas dos claves K_1 y K_2 es posible encontrar una clave K_3 tal que $E_{K_1} \circ E_{K_2} = E_{K_3}$ ($E_{K_1}(E_{K_2}(M)) = E_{K_3}(M)$ para cualquier $M \in \mathcal{M}$).

Un sistema criptográfico que cumpliera esta propiedad se denominaría sistema criptográfico cerrado.

Por otra parte, un sistema criptográfico puro sería uno en el que para tres claves K_1, K_2, K_3 existe una cuarta clave tal que $E_{K_1} \circ E_{K_2}^{-1} \circ E_{K_3} = E_{K_4}$.

Actualmente se sabe que DES no tiene ninguna estructura de grupo, aunque eso no impide que pudiera existir un trío de claves K_1, K_2 y K_3 que cumpliera la propiedad $E_{K_1} \circ E_{K_2} = E_{K_3}$. Sin embargo, se ha comprobado (Campbell y Wiener) que el cardinal del subgrupo de $S_{2^{64}}$ generado por las transformaciones que son cifrados del DES tiene un cardinal por encima de $1'94 \cdot 10^{2499}$.

Vamos por último a comentar algunos ataques e intentos de criptoanálisis al DES.

El primer ataque, y más sencillo es el *ataque por fuerza bruta*. En este se parte de uno o más textos cifrados, y se trata de encontrar la clave y el texto claro probando con todas las posibles claves hasta encontrar la correcta. El número de claves es $2^{56} = 72057594037927936 \sim 7'2 \cdot 10^{16}$. Si suponemos que encontramos la clave después de probar aproximadamente con la mitad de ellas, necesitaremos probar aproximadamente con 2^{55} claves.

Un segundo ataque es el *ataque con texto claro conocido*. En este caso, el criptoanalista tiene acceso a diferentes parejas de texto claro y su correspondiente cifrado, y ellas deben proporcionarle la información necesaria para encontrar la clave. Sin embargo, no se ha encontrado ningún método para romper el DES con este tipo de ataque.

También se puede intentar romper el DES con *ataque con texto claro elegido*. Ahora el criptoanalista puede elegir los textos claros y obtener los correspondientes textos cifrados. Aquí el criptoanalista dispone de más información, por la posibilidad de elegir él los textos claros. Sin embargo, hasta principios de los 90 no se conocía ningún método para romper el DES utilizando un número de estas parejas (texto claro, texto cifrado) menor a las pruebas necesarias para un ataque por fuerza bruta. Como ya comentamos, los israelíes Bihman y Shamir introdujeron el criptoanálisis diferencial, aunque los investigadores de IBM decían ya conocer dicho criptoanálisis cuando se publicó el DES. Se basa fundamentalmente en la comparación del OR exclusivo de parejas de textos en claro escogidos con el OR exclusivo de parejas de textos cifrados. Sin embargo, para las 16 rondas de que consta el DES este método se ha mostrado ineficiente.

También se ha intentado criptoanalizar el DES con el criptoanálisis lineal, introducido en 1993 por Matsui. Aquí se trata de encontrar relaciones lineales que representan una cierta probabilidad de relación existente entre algunos bits del mensaje en claro y otros del mensaje cifrado. Esto permite identificar los bits de la clave. Con este criptoanálisis se consiguió romper el DES de 16 rondas en 50 días mediante el análisis de 2^{43} textos en claro seleccionados, y utilizando 12 estaciones de trabajo. Sin embargo, la posibilidad de encontrar claves del DES en tiempo real aplicando criptoanálisis lineal es todavía teórica, aunque cabe la posibilidad de que agencias de inteligencia tengan los recursos necesarios para hacerlo.

En vista de esto, en lugar de utilizar el DES en su versión más simple, se ha propuesto utilizar un cifrado doble o triple, con varias claves, para así aumentar la seguridad.

1.4 *Triple DES*

Entre las mejoras propuesta, la más extendida es el Triple DES, que representaremos por TDES o 3DES. Para esto elegimos dos claves K_1 y K_2. La función de cifrado es entonces

$$E_{K_1} \circ E_{K_2}^{-1} \circ E_{K_1}$$

Para descifrar basta utilizar la función

$$E_{K_1}^{-1} \circ E_{K_2} \circ E_{K_1}^{-1}.$$

De esta forma obtenemos un método de cifrado diferente, pero lo fundamental es que elevamos de 2^{56} a 2^{112} el número de posibles claves, y en consecuencia incrementamos la dificultad de romper el criptosistema por el método de "*fuerza bruta*".
En la línea de mejorar el DES se pueden hacer variaciones como las que indicamos a continuación:

1. Modificación de las subclaves;
2. Dividir el bloque a cifrar en más de dos subbloques;
3. Modificación de las S-cajas;
4. Reiteración del DES en la línea del TDES.

Este algoritmo es un algoritmo robusto. El aumento del tamaño de la clave hace totalmente impensable un ataque por fuerza bruta, y todos los análisis que se han hecho al DES valen ahora para el TDES. Por tanto, si la seguridad fuera la única consideración para elegir un algoritmo de cifrado, el TDES sería una buena elección para las próximas décadas.
Pero el TDES tiene otros inconvenientes. En primer lugar es relativamente lento, especialmente para su implementación en software. El DES se pensó para implementación en hardware, pero su implementación en software no es muy eficiente. El TDES emplea el triple de pasos que el DES, y por tanto es tres veces más lento. Además el tamaño de los bloques, tanto del DES como del TDES es muy pequeño. Por razones, tanto de eficiencia como de seguridad, es mejor utilizar tamaños de bloque mayores.
Por eso, el TDES no es un candidato a perdurar durante mucho tiempo, y para ello el NIST realizó un concurso en 1997 para desarrollar un estándar de cifrado tan robusto como el TDES, pero más eficiente.

CRIPTOGRAFÍA VII

Advanced Encryption Standard (AES)

..... 1

Introducción

En enero de 1997, el National Institute of Standards and Technology (NIST), sustituto de la antigua NBS, por los motivos que acabamos de explicar, convocó a concurso público la adjudicación de un nuevo estándar de cifrado. Este nuevo estándar se denominaría AES. En la convocatoria se exigían unos requisitos míminos:

1. El algoritmo debe ser *simétrico de clave secreta.*
2. El algoritmo debe ser un *algoritmo de bloque.*
3. El algoritmo debe ser capaz de soportar las combinaciones clave-bloque de tamaños 128–128, 192–128 y 256–128.
4. El algoritmo debe poder ser fácilmente implementado tanto en hardware como en software.

Los criterios de evaluación de los proyectos presentados fueron los siguientes:

Seguridad. El factor más importante en la evaluación de los candidatos.

Coste.

1. El algoritmo debe ser accesible a todo el mundo y de libre distribución.
2. El algoritmo debe ser computacionalmente eficiente tanto en hardware como en software.
3. El algoritmo debe utilizar la menor memoria posible tanto en hardware como en software.

Características de implementación del algoritmo.

1. El algoritmo debe ser fácilmente implementable en distintas plataformas tanto en hardware como en software.

2. El algoritmo debe acomodarse a diferentes combinaciones clave-bloque además de las mínimas requeridas.

3. El algoritmo debe ser de diseño simple.

En esta ocasión, el concurso estaba abierto a todo el mundo, y el proceso de selección iba a ser totalmente transparente. Además, cualquiera podía evaluar las diferentes propuestas presentadas. El NIST publicaría las especificaciones técnicas de cada uno de los algoritmos presentados. Se pretendía de esta forma evitar cualquier tipo de polémica como la ocurrida años antes con el DES.

En mayo de 1998 se cerró el periodo de aceptación de propuestas. En total, se presentaron quince candidatos, que fueron presentados en una conferencia de candidatos en agosto de 1998 en California. Los quince candidatos son:

CAST-256, Entust Technologies, Inc. (C. Adams).
CRYPTON, Future Systems, Inc. (Chae Hoon Lim).
DEAL, L. Knudsen, R. Outerbridge.
DFC, CNRS-Ecole Normale Superiere (S. Vaudenay).
E2, NTT Nippon Telegraph and Telephone Corporation (M. Kanda).
FROG, TecApro International S.A. (D. Georgoudis, Leroux, Chaves).
HPC, R. Schoeppel.
LOKI97, L. Brown, J. Pieprzyk, J.Seberry.
MAGENTA, Deutshe Telekom AG (K. Huber).
MARS, IBM (N. Zunic).
RC6, RSA Laboratories (Rivest, M. Robshaw, Sidney, Yin).
RIJNDAEL, J. Daemen, V. Rijmen.
SAFER+, Cylink Corporation (L. Chen).
SERPENT, R. Anderson, E. Biham, L. Knudsen.
TWOFISH, B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson.

Esta presentación fue el inicio de una ronda de evaluación por parte de los criptógrafos de todo el mundo. En marzo de 1999, en Roma, se discutieron los resultados de los análisis efectuados, y cinco de los quince candidatos fueron rechazados. En agosto del mismo año, el NIST eligió a cinco candidatos finalistas. Estos eran MARS, RIJNDAEL, RC6, SERPENT y TWOFISH. Por último, en abril de 2000 se celebró en Nueva York la última conferencia, en la que la organización solicitó una valoración de los cinco finalistas a través de un cuestionario. El cifrado preferido fue el RIJNDAEL. El anuncio oficial del vencedor tuvo lugar el día 2 de agosto de 2000.

Esta decisión únicamente obliga a adoptar el nuevo criptosistema a la administración federal estadounidense, y en lo concerniente a información no clasificada. Pero la recomendación por parte del NIST es un aval de garantía para empresas y organizaciones. De esta forma, el AES va a ser probablemente el sistema de cifrado que más se usará en los próximos años. De hecho, la NSA aprobó en junio de 2003 el uso de AES para cifrar información clasificada: la secreta con claves de 128 bits, y la de alto secreto con claves de 192 y 256 bits.

. 2

Descripción del AES

Vamos a continuación a describir en detalle el algoritmo de cifrado AES. Como hemos dicho, AES cifra bloques de 128 bits, y devuelve bloque del mismo tamaño.
La unidad básica en el proceso del AES es el byte. Un bloque está formado entonces por 16 bytes. Estos bytes se distribuyen en una matriz 4×4 siguiendo el orden que nos muestra el siguiente diagrama:

$b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7$	$b_{32}b_{33}b_{34}b_{35}b_{36}b_{37}b_{38}b_{39}$	$b_{64}b_{65}b_{66}b_{67}b_{68}b_{69}b_{70}b_{71}$	$b_{96}\ b_{97}\ b_{98}\ b_{99}\ b_{100}b_{101}b_{103}b_{103}$
$b_8\ b_9\ b_{10}b_{11}b_{12}b_{13}b_{14}b_{15}$	$b_{40}b_{41}b_{42}b_{43}b_{44}b_{45}b_{46}b_{47}$	$b_{72}b_{73}b_{74}b_{75}b_{76}b_{77}b_{78}b_{79}$	$b_{104}b_{105}b_{106}b_{107}b_{108}b_{109}b_{110}b_{111}$
$b_{16}b_{17}b_{18}b_{19}b_{20}b_{21}b_{22}b_{23}$	$b_{48}b_{49}b_{50}b_{51}b_{52}b_{53}b_{54}b_{55}$	$b_{80}b_{81}b_{82}b_{83}b_{84}b_{85}b_{86}b_{87}$	$b_{112}b_{113}b_{114}b_{115}b_{116}b_{117}b_{118}b_{119}$
$b_{24}b_{25}b_{26}b_{27}b_{28}b_{29}b_{30}b_{31}$	$b_{56}b_{57}b_{58}b_{59}b_{60}b_{61}b_{62}b_{63}$	$b_{88}b_{89}b_{90}b_{91}b_{92}b_{93}b_{94}b_{95}$	$b_{120}b_{121}b_{122}b_{123}b_{124}b_{125}b_{126}b_{127}$

Cada byte se suele representar como 2 cifras hexadecimales, y se corresponde con un elemento del cuerpo $\mathbb{F}_{256}$ (ver apéndice al final de la sección).
Por ejemplo, a partir del siguiente bloque de 128 bits:

01001011110101101001011111000010101001010011011010110000101001010111111101010101101000011110101101000001101001010010100001011010

tendríamos la matriz

01001011	10100101	01111111	01000001
11010110	00110110	01010101	10100101
10010111	10110000	10100001	00101000
11000010	10100101	11101011	01011010

o en notación hexadecimal

4B	A5	7F	41
D6	36	55	A5
97	B0	A1	28
C2	A5	EB	5A

El algoritmo realiza una serie de transformaciones sobre el bloque. Las distintas etapas por las que va pasando el bloque se denominan *estado*.

. . . . . 2.1

Proceso de cifrado

En el proceso de cifrado hay cuatro transformaciones que se aplican varias veces. Éstas son Subbytes, Shiftrows, Mixcolumns, Addroundkey. Vamos a continuación a describir cada una de ellas:

Subbytes

Esta es una transformación que opera a nivel de bytes, y que podemos dividir en dos etapas. En primer lugar, calcula el inverso de cada elemento, y en segundo lugar se le aplica una transformación afín.

Inverso Recordemos que un byte puede ser visto como un elemento de $\mathbb{F}_{256}$, y que para representar $\mathbb{F}_{256}$ utilizamos el polinomio $x^8 + x^4 + x^3 + x + 1$ (que es irreducible en $\mathbb{Z}_2[x]$).

En tal caso, por ejemplo el byte 01101000 se corresponde con la clase del polinomio $x^6 + x^5 + x^3$.

Por ser $\mathbb{F}_{256}$ un cuerpo, todo elemento distinto de cero tiene inverso para el producto. La primera etapa de la transformación subbyte consiste en sustituir cada byte (elemento de $\mathbb{F}_{256}$) por su inverso multiplicativo. En esta etapa, el byte 00000000 se sustituye por él mismo.

En la sección del final del capítulo se explica como calcular el inverso de cada elemento.

Transformación Afín Dado un byte $b_7b_6b_5b_4b_3b_2b_1b_0$, la segunda etapa de la transformación subbyte lo sustituye por un nuevo byte $b'_7b'_6b'_5b'_4b'_3b'_2b'_1b'_0$. La relación entre ambos viene dada por:

$$\begin{pmatrix} b'_7 \\ b'_6 \\ b'_5 \\ b'_4 \\ b'_3 \\ b'_2 \\ b'_1 \\ b'_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

donde las operaciones están hechas en $\mathbb{Z}_2$.

Esta última transformación podemos escribirla como

$$b'_i = b_i \oplus b_{(i+4)\text{mód } 8} \oplus b_{(i+5)\text{mód } 8} \oplus b_{(i+6)\text{mód } 8} \oplus b_{(i+7)\text{mód } 8} \oplus c_i$$

donde c_i corresponde con el bit que ocupa la posición i del byte que en hexadecimal es 63, es decir, $c_7c_6c_5c_4c_3c_2c_1c_0 = 01100011 = 63$.

Veamos algún ejemplo.

El byte 00 en la primera etapa queda igual, y en la segunda queda:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Es decir, el transformado del byte 00 por la transformación subbytes es 63.

Vamos a calcular ahora como quedaría el byte 48 después de aplicarle la transformación *Subbytes*:

$$48^{-1} = Alog(FF - Log_{x+1}(48)) = Alog(FF - 13) = Alog(EC) = A7$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Es decir, el transformado de 48 es 52.

Los transformados de todos los elementos de $\mathbb{F}_{256}$ los podemos ver en la siguiente tabla:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Shiftrows

Esta transformación opera sobre las filas de un estado.
La primera fila del estado la deja tal cual.
La segunda fila la desplaza cíclicamente una posición hacia la izquierda.
La tercera fila la desplaza cíclicamente dos posiciones hacia la izquierda.
La cuarta fila la desplaza cíclicamente tres posiciones hacia la izquierda.
Así, el estado

4B	A5	7F	41
D6	36	55	A5
97	B0	A1	28
C2	A5	EB	5A

quedaría, después de aplicarle la transformación *Shiftrows*

4B	A5	7F	41
36	55	A5	D6
A1	28	97	B0
5A	C2	A5	EB

Mixcolumns

Esta transformación sobre las columnas de un estado. Una columna es vista como una matriz 4×1 con coeficientes en $\mathbb{F}_{256}$. El resultado de aplicar la transformación *Mixcolumns* a un estado es el resultado de multiplicar cada columna por la siguiente matriz 4×4 (con coeficientes en $\mathbb{F}_{256}$)

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

Por ejemplo, si tomamos la primera columna del ejemplo anterior tenemos:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} 4B \\ 36 \\ A1 \\ 5A \end{pmatrix} = \begin{pmatrix} 02 \cdot 4B + 03 \cdot 36 + 01 \cdot A1 + 01 \cdot 5A \\ 01 \cdot 4B + 02 \cdot 36 + 03 \cdot A1 + 01 \cdot 5A \\ 01 \cdot 4B + 01 \cdot 36 + 02 \cdot A1 + 03 \cdot 5A \\ 03 \cdot 4B + 01 \cdot 36 + 01 \cdot A1 + 02 \cdot 5A \end{pmatrix} =$$

$$= \begin{pmatrix} 96 + 5A + A1 + 5A \\ 4B + 6C + F8 + 5A \\ 4B + 36 + 59 + EE \\ DD + 36 + A1 + B4 \end{pmatrix} = \begin{pmatrix} 37 \\ 85 \\ CA \\ FE \end{pmatrix}$$

Nota:
Podría haberse definido la transformación *Mixcolumn* considerando cada columna de un estado como un elemento de $\mathbb{F}_{256}[x]_{x^4+1}$.
En ese caso, la transformación consiste únicamente en multiplicar por $03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$.
Una columna, como por ejemplo

4B
36
A1
5A

se correspondería con el polinomio $5A \cdot x^3 + A1 \cdot x^2 + 36 \cdot 01 + 4B$.
Entonces, se haría la multiplicación $(03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02) \cdot (5A \cdot x^3 + A1 \cdot x^2 + 36 \cdot 01 + 4B)$, y el resultado, que es un polinomio de grado 6, dividirlo por $x^4 + 1$ (o $01 \cdot x^4 + 01$). Los coeficientes del resto nos determinaría el resultado de aplicar la transformación *Mixcolumn*.
como la multiplicación, en $\mathbb{F}_{256}[x]_{x^4+1}$

Addroundkey

Esta transformación opera a nivel de bits. Para realizar esta tranformación necesitamos, cada vez que se aplica, una clave (denominada clave de ronda), que es del mismo tamaño que el estado. La transformación consiste en realizar un XOR (suma en $\mathbb{Z}_2$) bit a bit entre los bits del estado y los bits de la clave. Por ejemplo, si tenemos el estado

A2	C5	11	35
82	5F	90	0A
BE	C2	5D	B1
4C	73	26	AF

y la clave de ronda es

17	3A	45	22
BE	3A	0D	5B
B3	AA	60	19
C8	72	2F	BB

el resultado es (pasamos antes a notación binaria)

10100010	11000101	00010001	00110101
10000010	01011111	10010000	00001010
10111110	11000010	01011101	10110001
01001100	01110011	00100110	01011111

$\oplus$

00010111	00111010	01000101	00100010
10111110	00111010	00001101	01011011
10110011	10101010	01100000	00011001
11001000	01110010	00101111	10111011

$=$

10110101	11111111	01010100	00010111
00111100	01100101	10011101	01010001
00001101	01101000	00111101	10101000
10000100	00000001	00001001	11100100

Es decir:

A2	C5	11	35
82	5F	90	0A
BE	C2	5D	B1
4C	73	26	AF

$\oplus$

17	3A	45	22
BE	3A	0D	5B
B3	AA	60	19
C8	72	2F	BB

$=$

B5	FF	54	17
3C	65	9D	51
0D	68	3D	A8
84	01	09	E4

Una vez descritas las cuatro transformaciones básicas vamos a pasar a explicar como trabaja el algoritmo de cifrado AES.

Para esto, vamos a denotar por N_b al número de bits de un bloque dividido por 32 (es decir, el número de columnas de un bloque, pues cada columna está formada por 4 bytes, es decir, 32 bits) y N_k el número de bits de la clave dividido por 32 (por tanto N_k puede tomar los valores 4, 6 u 8, que se corresponden con los tamaños de claves de 128, 192 o 256 bits respectivamente).

El algoritmo realiza una serie de rondas, cuyo número depende del valor de N_k. Para $N_k = 4$ realiza 10 rondas, para $N_k = 6$ realiza 12 rondas y para $N_k = 8$ realiza 14 rondas. Vamos a denotar por N_r al número de rondas. Cada ronda consiste en la realización de las cuatro transformaciones que acabamos de explicar, en el orden que hemos puesto, salvo la última,

en la que se suprime la transformación *Mixcolumn*. Antes de realizar la primera ronda, se realiza una transformación *Addroundkey*
Por tanto, el algoritmo quedaría como sigue:

Addroundkey	Transformación Inicial
Subbytes Shiftrows Mixcolumns Addroundkey	$N_r - 1$ Rondas
Subbytes Shiftrows Addroundkey	Ronda final

Como vemos, la transformación *Addroundkey* se aplica $N_r + 1$ veces. Para cada una de las veces, necesitamos una clave de ronda. Estas claves se obtienen a partir de la clave inicial a partir de un proceso denominado *Expansión de clave*.

3

Expansión de clave

Dada la clave inicial K, ésta se agrupa igual a como vimos al inicio con el bloque a cifrar. El número de columnas es N_k. Puesto que cada clave de ronda tiene 4 columnas (el mismo tamaño del estado) entonces necesitamos un total de $4 \cdot (N_r + 1)$ columnas (es decir, 44, 52 ó 60). Cada una de las columnas la vamos a denotar por W_i, con i variando entre 0 y $4 \cdot N_r + 3$. Inicialmente partimos de las columnas $W_0, W_1, \cdots, W_{N_k-1}$, y a partir de ellas debemos generar las columnas $W_{N_k}, W_{N_k+1}, \cdots, W_{4 \cdot N_r+3}$.
Vamos a ver como se construye la columna (palabra) W_i a partir de las anteriores. Para ello, se distinguen dos casos, dependiendo del valor de N_k, y a su vez en ellos distinguiremos varios casos en función de i.
Antes de comenzar, definimos, para $i = 1, 2, \cdots$ el vector Rcon[i] (*Round constant*). Este vector está formado por 4 bytes, y es:

02^{i-1}
00
00
00

Es decir, el primer byte es el resultado de calcular la potencia x^{i-1} en $\mathbb{F}_{256}$. Los distintos valores son entonces:

$$01 \quad 02 \quad 04 \quad 08 \quad 10 \quad 20 \quad 40 \quad 80 \quad 1B \quad 36 \quad \cdots$$

Los tres bytes restantes son nulos.

– $N_k = 4, 6$

Vamos a partir de la clave

F2	7A	59	73
C2	96	35	59
95	B9	80	F6
F2	43	7A	7F

y a partir de ella vamos a ver como se van obteniendo las distintas subclaves.

1. Generación de W_i cuando $i \equiv 0 (\text{mód } N_k)$.

   En este caso se realizan tres operaciones:

   a) Se toma la palabra W_{i-1} y se le aplica la transformación *Rotword*.
   Esta transformación consiste en desplazar cíclicamente hacia arriba la palabra en cuestión. En el caso que nos ocupa, si queremos calcular W_4 haríamos:

$$W_3 = \begin{array}{|c|}\hline 73 \\ \hline 59 \\ \hline F6 \\ \hline 7F \\ \hline \end{array} \xrightarrow{\text{Rotword}} \begin{array}{|c|}\hline 59 \\ \hline F6 \\ \hline 7F \\ \hline 73 \\ \hline \end{array}$$

   b) Al resultado del paso anterior se le aplica la transformación Subbytes (ahora se denomina *Subword*)

$$\begin{array}{|c|}\hline 59 \\ \hline F6 \\ \hline 7F \\ \hline 73 \\ \hline \end{array} \xrightarrow{\text{Subword}} \begin{array}{|c|}\hline CB \\ \hline 42 \\ \hline D2 \\ \hline 8F \\ \hline \end{array}$$

   c) Se hace un XOR con el resultado obtenido en el paso anterior, W_{i-N_k} y $\text{Rcon}[i/N_k]$. En nuestro caso $i - N_k = 0$ e $i/N_k = 1$. Por tanto:

$$\begin{array}{ccccc} W_0 & & \text{Rcon}[1] & & W_4 \\ \begin{array}{|c|}\hline F2 \\ \hline C2 \\ \hline 95 \\ \hline F2 \\ \hline \end{array} & \oplus & \begin{array}{|c|}\hline 01 \\ \hline 00 \\ \hline 00 \\ \hline 00 \\ \hline \end{array} & \oplus & \begin{array}{|c|}\hline CB \\ \hline 42 \\ \hline D2 \\ \hline 8F \\ \hline \end{array} = \begin{array}{|c|}\hline 38 \\ \hline 80 \\ \hline 47 \\ \hline 7D \\ \hline \end{array} \end{array}$$

2. Generación de W_i cuando $i \not\equiv 0 (\text{mód } N_k)$.

   En este caso es mucho más sencillo. Basta hacer un XOR entre W_{i-1} y W_{i-N_k}.

W_1 W_4 W_5 W_2 W_5 W_6

7A		38	42		59		42	1B
96	$\oplus$	80	=	16		35	$\oplus$	16	=	23
B9		47		FE		80		FE		7E
43		7D		3E		7A		3E		44

– $N_k = 8$

Aquí vamos a trabajar con la clave inicial

9B	8E	A5	20	A8	93	BE	B7
A3	69	1A	67	B0	D1	49	5D
54	25	8B	FC	9C	94	84	5B
11	AF	5F	DE	1A	CD	6E	9A

1. Generación de W_i cuando $i \equiv 0 (\text{mód } 8)$

   Se hace igual que el caso anterior. Se toma W_{i-1}, se le aplica la transformación *Rotword*, la transformación *Subbytes* y se hace un XOR con $W_{i-8} \oplus \text{Rcon}[i/8]$. Por ejemplo, vamos a calcular W_8.

W_7

B7		5D		4C
5D	Rotword $\longrightarrow$	5B	Subword $\longrightarrow$	39
5B		9A		B8
9A		B7		A9

W_0 Rcon[1] W_8

9B		01		4C		D6
A3	$\oplus$	00	$\oplus$	39	=	9A
54		00		B8		EC
11		00		A9		B8

2. Generación de W_i cuando $i \not\equiv 0 (\text{mód } 8)$ e $i \not\equiv 4 (\text{mód } 8)$

   También aquí es como en el caso anterior: $W_i = W_{i-1} \oplus W_{i-8}$.

W_1 W_8 W_9 W_2 W_9 W_{10} W_3 W_{10} W_{11}

8E		D6		58		A5		58		FD		20		FD		ED
69	$\oplus$	9A	=	F3		1A	$\oplus$	7F	=	E9		67	$\oplus$	E9	=	8E
25		EC		C9		8B		C9		42		FC		42		BE
AF		B8		17		5F		17		48		DE		48		96

3. Generación de W_i cuando $i \equiv 4 (\text{mód } 8)$.

   En este caso se realiza en dos etapas:

   a) Se aplica la transformación *Subword* a W_{i-1}.

$$W_{11}$$

$$\begin{array}{|c|} \hline ED \\ \hline 8E \\ \hline BE \\ \hline 96 \\ \hline \end{array} \xrightarrow{\text{Subword}} \begin{array}{|c|} \hline 55 \\ \hline 19 \\ \hline AE \\ \hline 90 \\ \hline \end{array}$$

   b) Se realiza un XOR con W_{i-8}.

$$\begin{array}{c} W_4 \\ \begin{array}{|c|} \hline A8 \\ \hline B0 \\ \hline 9C \\ \hline 1A \\ \hline \end{array} \end{array} \oplus \begin{array}{c} \\ \begin{array}{|c|} \hline 55 \\ \hline 19 \\ \hline AE \\ \hline 90 \\ \hline \end{array} \end{array} = \begin{array}{c} W_{12} \\ \begin{array}{|c|} \hline FD \\ \hline A9 \\ \hline 32 \\ \hline 8A \\ \hline \end{array} \end{array}$$

De esta forma obtenemos las columnas W_i, $0 \leq i \leq 4 \cdot N_r + 3$ (las primeras N_b columnas son las de la clave inicial).
Con estas columnas formamos las claves K_j, $0 \leq j \leq N_r$. Cada clave K_j está formada por las columnas W_{4j}, W_{4j+1}, W_{4j+2} y W_{4j+3}. La clave K_0 se utiliza en la transformación inicial, las claves $K_1, \cdots, K_{N_r-1}$ en las $r-1$ rondas centrales, y por último, la clave K_{N_r} en la ronda final.
Vamos a ver, por último como es el proceso de descifrado.

4 Proceso de descifrado

Para descifrar un bloque que haya sido cifrado con el algoritmo AES deberemos invertir todas las transformaciones que se realizan para cifrar, y el orden en que las aplicamos. Veamos la inversa de cada una de las transformaciones básicas:

4.1 *Invsubbytes*

A partir de la tabla que construimos para la transformación *Subbytes* es fácil construir la tabla para la transformación inversa. Por ejemplo, si el transformado del byte D9 es 35, entonces en la tabla inversa el 35 debe transformarse en el D9. Haciendo esto con los 256 elementos de $\mathbb{F}_{256}$ obtenemos la siguiente tabla:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

. . . . . 4.2

InvShiftrows

En este caso, la función inversa consiste en hacer desplazamientos circulares 1 2 y 3 posiciones pero hacia la derecha.

. . . . . 4.3

InvMixcolumn

Si la transformación *Mixcolumn* era multiplicar por una matriz, su transformación inversa es multiplicar por la matriz inversa. La matriz inversa es:

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix}$$

Llamaremos a esta matriz IM.

También podría haberse obtenido calculando el inverso del polinomio $03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$ en $\mathbb{F}_{256}[x]_{x^4+1}$. Este último anillo no es un cuerpo, pues $x^4 + 1$ es reducible, pero como $mcd(x^4 + 1, 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02) = 1$ entonces si podemos calcular el inverso de ese polinomio. El inverso resulta ser $0B \cdot x^3 + 0D \cdot x^2 + 09 \cdot x + 0E$.

. . . . . 4.4

InvAddroundkey

Puesto que la transformación *Addroundkey* es hacer un XOR con la clave de ronda, su inversa es ella misma.
Por tanto, el proceso de descifrado quedaría:

Addroundkey InvShiftrows InvSubbytes	Ronda inicial
Addroundkey InvMixcolumns InvShiftrows InvSubbytes	$N_r - 1$ Rondas
Addroundkey	Transformación Final

Donde en la ronda inicial se utiliza la clave K_{N_r}; las claves $K_{N_r-1}, \cdots K_2, K_1$ (en este orden) en las rondas intermedias; y en la transformación final la clave K_0.
Ahora podemos incluir las transformaciones *InvShiftrows* e *InvSubbytes* de la ronda inicial en las rondas centrales. De esta forma, cada una de las rondas intermedias seguiría la secuencia *InvShiftrows*, *InvSubbytes*, *Addroundkey*, *InvMixcolumns*. Por tanto, las dos últimas transformaciones *InvShiftrows* e *InvSubbytes* de la última ronda central pasarían a la ronda final.
Por otra parte, podemos ver que las transformaciones *InvShiftrows* e *InvSubbytes* conmutan, luego podemos ponerlas en orden inverso. De esta forma, el proceso de descifrado queda:

Addroundkey	Transformación Inicial
InvSubbytes InvShiftrows Addroundkey InvMixcolumns	$N_r - 1$ Rondas
InvSubbytes InvShiftrows Addroundkey	Ronda final

Por último, si partimos de un estado S, y le aplicamos la transformación *Addroundkey* con la clave K_i el resultado es $S \oplus K_i$. Si ahora le aplicamos la transformación *InvMixcolumns* el

resultado es $IM(S \oplus K_i) = IM(S) \oplus IM(K_i)$. Podemos ver que esto es lo mismo que aplicar la transformación *InvMixcolumns* ($IM(S)$) y posteriormente la transformación *Addroundkey*, pero con clave $IM(K_i)$. Vamos a denotar a esta transformación por *AddroundIMkey*. En tal caso, podríamos escribir el proceso de descifrado como sigue:

Addroundkey	Transformación Inicial
InvSubbytes InvShiftrows InvMixcolumns AddroundIMkey	$N_r - 1$ Rondas
InvSubbytes InvShiftrows Addroundkey	Ronda final

Y vemos que tiene una estructura similar al proceso de cifrado.
El algoritmo que aquí se ha explicado se corresponde con el estándar descrito por el NIST. Posteriormente se ha añadido más versatilidad, dando posibilidades de variar, no sólo el tamaño de la clave sino también el tamaño del bloque, permitiendo bloques de 128, 192 y 256 bits. La siguiente tabla nos muestra el número de rondas en función del tamaño de bloque y el tamaño de clave (recordemos que N_b denota el número de bits de un bloque dividido por 32):

N_r	$N_b = 4$	$N_b = 6$	$N_b = 8$
$N_k = 4$	10	12	14
$N_k = 6$	12	12	14
$N_k = 8$	14	14	14

Esto añade algunas ligeras modificaciones al algoritmo, que vamos a ver a continuación.
En primer lugar, a la hora de generar las claves es necesario obtener $N_b \cdot (N_r + 1)$ palabras. La forma de generarlas es igual a la estudiada anteriormente.
La transformación *Subbytes* se mantiene igual que antes.
El número de posiciones que se desplaza cada fila en la transformación *Shiftrows* cambia cuando $N_b = 8$. En la primera fina no hay desplazamientos, en la segunda el desplazamiento es de una posición (hasta ahora no hay ninguna variación), en la tercera tres posiciones y en la cuarta cuatro.
Por último, la transformación *Mixcolumn* no sufre ninguna variación.

5

Apéndice: Cuerpos finitos. El cuerpo $\mathbb{F}_{256}$.

Vamos a dar en este apéndice unas pinceladas sobre cuerpos finitos, y posteriormente nos centraremos en la estructura del cuerpo de cardinal 256, así como en la forma de representar sus elementos.
Comenzamos recordando qué es un cuerpo.
Dado un conjunto K, diremos que tiene estructura de cuerpo si en él tenemos definidas dos operaciones binarias, que llamaremos suma y producto y representaremos como $+$ y $\cdot$, y que satisfacen las siguientes propiedades:

Suma

1. **Asociativa**: Para cualesquiera $a, b, c \in K$ se tiene que $a + (b + c) = (a + b) + c$.
2. **Conmutativa**: Para cualesquiera $a, b \in K$ se tiene que $a + b = b + a$.
3. **Elemento neutro**: Existe un elemento en K, que denominaremos 0 tal que para cualquier $a \in K$ se tiene que $a + 0 = a$.
4. **Opuesto**: Para cualquier $a \in K$, existe $b \in K$ tal que $a + b = 0$. A este elemento lo denominaremos $-a$.

Dados $a, b \in K$, denotaremos por $a - b$ al resultado de $a + (-b)$.

Producto

1. **Asociativa**: Para cualesquiera $a, b, c \in K$ se tiene que $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
2. **Conmutativa**: Para cualesquiera $a, b \in K$ se tiene que $a \cdot b = b \cdot a$.
3. **Elemento neutro**: Existe un elemento en K, que denominaremos 1 tal que para cualquier $a \in K$ se tiene que $a \cdot 1 = a$.
4. **Inverso**: Para cualquier $a \in K \setminus \{0\}$, existe $b \in K$ tal que $a \cdot b = 1$. A este elemento lo denominaremos a^{-1}.

Dados $a, b \in K$, con $b \neq 0$, denotaremos por a/b o $\frac{a}{b}$ al resultado de $a \cdot b^{-1}$.

Suma y producto

1. **Distributiva**: Para cualesquiera $a, b, c \in K$ se verifica que $(a + b) \cdot c = a \cdot c + b \cdot c$.

En resumen, un cuerpo podemos decir que es un conjunto en el que podemos sumar, restar, multiplicar y dividir (salvo por cero), con las propiedades usuales de esas operaciones.
Ejemplos de cuerpos son $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}_p$, con p un número primo.
No son cuerpos, por ejemplo $\mathbb{N}$ (no podemos restar ni dividir), $\mathbb{Z}$ (no podemos dividir), $\mathcal{M}_3(\mathbb{Q})$ (el producto no es conmutativo, y hay matrices que no tiene inversa), $\mathbb{Z}_n$, con n compuesto (si d es un divisor de n, distinto de 1 y n, entonces d no tiene inverso).

Consideramos el conjunto $X = \{0, 1, a, b\}$. Vamos a definir dos operaciones en X que van a darle a X estructura de cuerpo. Las operaciones las representamos en las siguientes tablas:

$+$	0	1	a	b
0	0	1	a	b
1	1	0	b	a
a	a	b	0	1
b	b	a	1	0

$\cdot$	0	1	a	b
0	0	0	0	0
1	0	1	a	b
a	0	a	b	1
b	0	b	1	a

Se puede comprobar sin mucha dificultad que X, con las operaciones definidas, tiene estructura de cuerpo.
Vamos a ver a continuación una forma para construir nuevos cuerpos. El proceso va a ser una generalización del que se sigue para construir $\mathbb{Z}_p$ a partir de $\mathbb{Z}$.

. . . . . 5.1

Polinomios con coeficientes en un cuerpo.

0. Generalidades
Supongamos que K es un cuerpo, y consideramos el conjunto de los polinomios (en una variable) con coeficientes en K, que llamaremos K[x].
Los elementos de K[x] los podemos representar como $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$, donde $n \in \mathbb{N}$, y $a_i \in K$, $a_n \neq 0$. A estos elementos, hay que añadir el polinomio 0, que en principio queda fuera de esta representación.
En el conjunto de los polinomios tenemos definidas dos operaciones, la suma y el producto, que suponemos conocida.
Las propiedades de estas operaciones las recordamos a continuación:

- La suma de polinomios es asociativa, es decir, $p(x) + (q(x) + r(x)) = p(x) + q(x)) + r(x)$. Nótese que esta propiedad es necesaria para poder definir el producto tal y como se ha hecho aquí.
- La suma de polinomios es conmutativa.
- La suma tiene un elemento neutro, que es el polinomio 0.
- Dado $p(x) \in A[x]$ existe $q(x) \in A[x]$ tal que $p(x) + q(x) = 0$. Denotaremos como $-p(x)$ a este polinomio.
- El producto de polinomios es asociativo y conmutativo.
- El producto tiene un elemento neutro, que es el polinomio 1.
- La suma es distributiva con respecto al producto.

Estas propiedades nos dicen que K[x] es un anillo conmutativo.
Además, podemos identificar K como los elementos de K[x] de la forma $p(x) = a$.

Ejemplo:
Sea $A = \mathbb{Q}$, y sean $p(x) = 2x^3 + 3x^2 + 7x + 9$ y $q(x) = 6x^2 + 5x + 4$. Entonces:

$$p(x) + q(x) = 2x^3 + (3+6)x^2 + (7+5)x + (9+4) = 2x^3 + 9x^2 + 12x + 13$$

$$\begin{aligned} p(x) \cdot q(x) &= p(x) \cdot (6x^2) + p(x) \cdot (5x) + p(x) \cdot 4 \\ &= (12x^5 + 18x^4 + 42x^3 + 54x^2) + (10x^4 + 15x^3 + 35x^2 + 45x) + (8x^3 + 12x^2 + 28x + 36) \\ &= 12x^5 + 28x^4 + 65x^3 + 101x^2 + 73x + 36 \end{aligned}$$

Normalmente, para efectuar la multiplicación dispondremos los datos de la siguiente forma:

$p(x)$			2	3	7	9
$q(x)$				6	5	4
$p(x) \cdot 4$			8	12	28	36
$p(x) \cdot 5x$		10	15	35	45	
$p(x) \cdot 6x^2$	12	18	42	54		
$p(x) \cdot q(x)$	12	28	65	101	73	36

luego el resultado final es $12x^5 + 28x^4 + 65x^3 + 101x^2 + 73x + 36$.
Recordamos a continuación algunos conceptos referentes a los polinomios:

Definición 1. Sea K un anillo conmutativo y $p(x) = a_n x^n + a_{n-1}x^{n-1} + a_1 x + a_0 \in K[x]$.

i) Si $a_n \neq 0$ entonces se dice que el polinomio $p(x)$ tiene **grado** n ($gr(p(x)) = n$). Nótese que no se ha definido el grado del polinomio 0. En ocasiones, consideraremos que el grado del polinomio 0 es -1.

ii) Al elemento $a_k \in A$ se le llama **coeficiente de grado k**, y a la expresión $a_k x^k$, **término de grado k**.

iii) El coeficiente de grado n de un polinomio de grado n se llama **coeficiente líder**, y a la expresión $a_n x^n$ **término líder**.

iv) El coeficiente de grado 0 de un polinomio se le llama **término independiente**.

v) Un polinomio cuyo coeficiente líder valga 1 se dice que es un **polinomio mónico**.

vi) Un polinomio que, bien tiene grado 0, o bien es el polinomio 0 se dice que es un **polinomio constante**.

Ejemplo:
Sean $p(x) = 3x^3 + 5x + 2$ y $q(x) = x^4 + 2x^3 + 3x^2 + 5x + 8$ dos polinomios con coeficientes en $\mathbb{Z}_{11}$. Entonces:
- $gr(p(x)) = 3$ y $gr(q(x)) = 4$.
- El coeficiente de grado 2 de $p(x)$ es 0, mientras que el coeficiente de grado 2 de $q(x)$ es 3. El coeficiente de grado 5 de $q(x)$ es cero.
- El coeficiente líder de $p(x)$ es 3, mientras que el coeficiente líder de $q(x)$ es 1. Por tanto, $q(x)$ es mónico, mientras que $p(x)$ no lo es.

- Los términos independientes de $p(x)$ y $q(x)$ son 2 y 8 respectivamente.
- Ninguno de los dos polinomios son constantes.

Proposición 5.1. Sean $p(x), q(x) \in K[x]$ dos polinomios distintos de cero. Entonces:

$$gr(p(x) + q(x)) \leq máx\{gr(p(x), q(x)\}$$

$$gr(p(x) \cdot q(x)) = gr(p(x)) + gr(q(x))$$

En general, para polinomios se tiene la relación $gr(p(x) \cdot q(x)) \leq gr(p(x)) + gr(q(x))$. Sin embargo cuando los coeficientes los tomamos en un cuerpo podemos asegurar que se tiene la igualdad
Veamos a continuación la evaluación de un polinomio en un punto.

Definición 2. Sea K un cuerpo, $p(x) = a_n x^n + \cdots + a_1 x + a_0 \in K[x]$ y $a \in K$. Se define la evaluación de $p(x)$ en el punto a, $Ev_a(p(x))$ como el elemento de K:

$$Ev_a(p(x)) = a_n a^n + \cdots + a_1 a + a_0$$

Dicho de otra forma, $Ev_a(p(x))$ es el resultado de sustituir en la expresión de $p(x)$ el símbolo x por a. De esta forma tenemos definida una aplicación (morfismo de anillos) $Ev_a : K[x] \to K$. Normalmente, escribiremos $p(a)$ en lugar de $Ev_a(p(x))$.

Proposición 5.2. Dado K un anillo y $p_1(x), p_2(x) \in K[x]$

1. Si $q(x) = p_1(x) + p_2(x)$ entonces $q(a) = p_1(a) + p_2(a)$ (es decir, $Ev_a(p_1(x) + p_2(x)) = Ev_a(p_1(x)) + Ev_a(p_2(x))$.
2. Si $q(x) = p_1(x) \cdot p_2(x)$ entonces $q(a) = p_1(a) \cdot p_2(a)$ (es decir, $Ev_a(p_1(x) \cdot p_2(x)) = Ev_a(p_1(x)) \cdot Ev_a(p_2(x))$.

Usando la aplicación evaluación, cada polinomio de $K[x]$ determina una aplicación $K \to K$, dada por $a \mapsto p(a)$.

Ejemplo:

1. El polinomio $x^3 + 3x^2 + 2x + 2 \in \mathbb{Z}_5[x]$ determina la aplicación $\mathbb{Z}_5 \to \mathbb{Z}_5$ siguiente:

$$0 \mapsto 2 \quad 1 \mapsto 3 \quad 2 \mapsto 1 \quad 3 \mapsto 2 \quad 4 \mapsto 2$$

2. El polinomio $x^2 + x + 1 \in \mathbb{Z}_2[x]$ determina la aplicación

$$0 \mapsto 1 \qquad 1 \mapsto 1$$

es decir, la aplicación constante 1.

Si el cuerpo K es infinito (por ejemplo, $K = \mathbb{Q},\ \mathbb{R},\ \mathbb{C}$) entonces dos polinomios distintos determinan siempre aplicaciones distintas. Si el número de elementos del cuerpo K es finito, entonces hay polinomios distintos que determinan la misma aplicación.
Una función $f : K \to K$ se dice que es polinómica si existe un polinomio $p(x) \in K[x]$ tal que la aplicación que determina es igual a f (es decir, f se puede representar mediante un polinomio)
Si K es infinito, hay funciones $K \to K$ que no son polinómicas (por ejemplo, la función $\mathbb{R} \to \mathbb{R}$ dada por $x \mapsto e^x$ no es polinómica). Si K es finito, toda función es polinómica, y hay infinitos polinomios que la representan.
Terminamos este apartado con el algoritmo de la división.

Teorema 5.1 (Algoritmo de la división). Sea K un cuerpo, y $p(x), q(x)$ dos polinomios de $K[x]$, con $q(x) \neq 0$. Entonces existen únicos polinomios $c(x), r(x) \in K[x]$ tales que:

$$p(x) = q(x) \cdot c(x) + r(x)$$

$$r(x) = 0 \text{ ó } gr(r(x)) < gr(q(x)).$$

Los polinomios $c(x)$ y $r(x)$ son llamados cociente y resto respectivamente.

Demostración:
Demostraremos la existencia de los polinomios. La unicidad se deja como ejercicio.
La demostración la haremos por inducción sobre el grado de $p(x)$. El caso $p(x) = 0$ queda fuera de esta demostración, pues no tiene grado; claro que para $p(x) = 0$ basta tomar $c(x) = r(x) = 0$.
Procedamos ya a la inducción. Sea $m = gr(q(x))$ y $n = gr(p(x))$.
Paso 1 Para $n = 0, 1, \cdots, m-1$ se tiene que $p(x) = q(x) \cdot 0 + p(x)$, y $gr(p(x)) < gr(q(x))$, luego ya está hecho.
Paso 2 Supongamos que el resultado es cierto para todo polinomio de grado menor que n (incluimos el polinomio 0). Si el coeficiente líder de $p(x)$ es a_n y el coeficiente líder de $q(x)$ es b_m, entonces se tiene que
$p(x) - a_n(b_m)^{-1}x^{n-m}q(x)$ es un polinomio de grado menor que n (¿por qué?), luego existen $c_1(x)$ y $r(x)$ tales que

$$p(x) - a_n(b_m)^{-1}x^{n-m}q(x) = q(x) \cdot c_1(x) + r(x)$$

$$gr(r(x)) < m \text{ ó } r(x) = 0.$$

Basta entonces tomar $c(x) = c_1(x) + a_n(b_m)^{-1}x^{n-m}$, y los polinomios $c(x)$ y $r(x)$ satisfacen las condiciones requeridas. ■

Ejemplo:
Calculemos el cociente y el resto de la división del polinomio $p(x) = 2x^4 + 3x^3 + 5x + 1$ entre $q(x) = 3x^3 + x + 6$ en $\mathbb{Z}_7[x]$. Lo haremos siguiendo los pasos hechos en la demostración precedente.

Notemos en primer lugar que $gr(p(x)) > gr(q(x))$.
Calculamos 3^{-1}. Se tiene que $3^{-1} = 5$.
Tomamos entonces el término $2 \cdot 5 \cdot x^{4-3} = 3x$.
Hallamos $p_1(x) = p(x) - 3xq(x) = p(x) + 4xq(x) = 3x^3 + 4x^2 + x + 1$.
Dado que $gr(p_1(x)) \geq gr(q(x))$ continuamos dividiendo. Tomamos el término $3 \cdot 5x^{3-3} = 1$
Hallamos $p_2(x) = p_1(x) - 1q(x) = p_1(x) + 6q(x) = 4x^2 + 2$.
Dado que $gr(p_2(x)) < gr(q(x))$ la división ha terminado. El cociente es $c(x) = 3x + 1$ y el resto $r(x) = 4x^2 + 2$.
Los cálculos podemos disponerlos como sigue:

2	3	0	5	1	|	3	0	1	6
5	0	4	3			3	1		
	3	4	1	1					
	4	0	6	1					
		4	0	2					

Podemos ahora ver que entre los polinomios con coeficientes en un cuerpo, y los números enteros tenemos muchas semejanzas.
En ambos tenemos definida una suma y un producto, con las mismas propiedades (asociativa, conmutativa y existencia de elemento neutro). Además cualquier elemento tiene un opuesto para la suma, lo que permite restar.
Y en ambos casos, dados dos elementos, tenemos una forma de obtener un cociente y un resto, de forma que el resto sea, de algún modo, menor que el divisor.

> **Nota:** Un anillo A conmutativo. Se dice que es un dominio euclídeo si en él tenemos definida una aplicación *grado*, $g : A^* \to \mathbb{N}$ satisfaciendo dos propiedades:
>
> - $g(ab) \geq g(a)$ para $b \neq 0$
> - Para todo $a, b \in A$, $b \neq 0$, existen $q, r \in A$ tales que $a = bq + r$ y $g(r) < g(a)$
>
> En $\mathbb{Z}$ la aplicación *grado* puede ser el valor absoluto, mientras que en $K[x]$ resulta ser el grado.

Definición 3. Sean $p(x), q(x) \in K[x]$, con $q(x) \neq 0$. Se definen los polinomios $p(x)$ mód $q(x)$ y $p(x)$ div $q(x)$ como el resto y el cociente de dividir $p(x)$ entre $q(x)$.
Cuando $p(x)$ mód $q(x) = 0$, denotaremos por $\frac{p(x)}{q(x)}$ al polinomio $p(x)$ div $q(x)$.

Ejemplo:

1. En $\mathbb{Z}_3[x]$, se verifica que:

$$x^5 + x^4 + 2x^3 + x^2 + x + 1 \text{ mód } x^2 + 2x + 1 = 2$$
$$x^5 + x^4 + 2x^3 + x^2 + x + 1 \text{ div } x^2 + 2x + 1 = x^3 + 2x^2 + 2.$$

2. En $\mathbb{Z}_5[x]$:

$$x^5 + x^4 + 2x^3 + x^2 + x + 1 \text{ mód } x^2 + 2x + 1 = 6x$$
$$x^5 + x^4 + 2x^3 + x^2 + x + 1 \text{ div } x^2 + 2x + 1 = x^3 + 4x^2 + 3x + 1.$$

El hecho de ser $K[x]$ un dominio euclídeo, al igual que $\mathbb{Z}$ permite que muchos resultados y conceptos que se tienen en los números enteros, se puedan trasladar a polinomios con coeficientes en un cuerpo.
Enumeramos a continuación algunos de estos conceptos, que explicaremos brevemente en lo que sigue:

1. Máximo común divisor y Mínimo común múltiplo.
2. Algoritmo de Euclides.
3. Identidad de Bezout.
4. Ecuaciones diofánticas.
5. Factorización única como producto de irreducibles (primos).
6. Congruencias.
7. Teorema chino del resto y sistemas de congruencias.
8. Cocientes.
9. Cálculo de inversos.

1. Máximo común divisor y Mínimo común múltiplo

Definición 4. Sea K un cuerpo, y $p(x), q(x) \in K[x]$. Se dice que $d(x) \in K[x]$ es un máximo común divisor de $p(x)$ y $q(x)$ si:

1. $d(x)|p(x)$ y $d(x)|q(x)$.
2. Si $c(x)|p(x)$ y $c(x)|q(x)$ entonces $c(x)|d(x)$.

Nota:

1. La primera condición de la definición nos dice que $d(x)$ debe ser un divisor común de $p(x)$ y $q(x)$. La segunda condición nos dice que este divisor común es el "más grande" de los divisores comunes.

2. Si $d(x)$ es un máximo común divisor de $p(x)$ y $q(x)$ y $a \in K^*$ entonces $a \cdot d(x)$ es también un máximo común divisor de $p(x)$ y $q(x)$. De hecho, cualquier polinomio que sea un máximo común divisor de $p(x)$ y $q(x)$ es de la forma $a \cdot d(x)$. De todos estos, hay uno, y sólo uno que es mónico. Denotaremos por $mcd(p(x), q(x))$ al único máximo común divisor de $p(x)$ y $q(x)$ que es mónico.

3. Aquí se ha definido el máximo común divisor de dos polinomios. Podría haberse definido de forma análoga el máximo común divisor de 3 ó más.

Se deja como ejercicio dar la definición de mínimo común múltiplo.
Veremos a continuación algunas propiedades referentes al máximo común divisor. Supongamos que tenemos $p(x), q(x), r(x), d(x) \in K[x]$, y supondremos que los cuatro polinomios son mónicos.

Propiedades:

1. $mcd(p(x), q(x)) = mcd(a \cdot p(x), q(x)) = mcd(p(x), a \cdot q(x))$, donde $a \in K^*$.
2. $mcd(p(x), 0) = p(x)$ y $mcd(p(x), 1) = 1$
3. Si $p(x)|q(x)$ entonces $mcd(p(x), q(x)) = p(x)$.
4. $mcd(p(x), mcd(q(x), r(x)) = mcd(mcd(p(x), q(x)), r(x)) = mcd(p(x), q(x), r(x))$.
5. $mcd(p(x) \cdot r(x), q(x) \cdot r(x)) = mcd(p(x), q(x)) \cdot r(x)$
6. Si $d(x)|p(x)$ y $d(x)|q(x)$ entonces $mcd\left(\frac{p(x)}{d(x)}, \frac{q(x)}{d(x)}\right) = \frac{mcd(p(x),q(x))}{d(x)}$.

Como ejercicio, se deja enunciar propiedades análogas para el mínimo común múltiplo.

2. Algoritmo de Euclides

Los siguientes resultados son análogos a los que se tienen para números enteros.

Lema 5.1. Sean $p(x), q(x) \in K[x]$. Entonces, para cualquier $c(x) \in K[x]$ se tiene que

$$mcd(p(x), q(x)) = mcd(q(x), p(x) - c(x)q(x)).$$

Corolario 5.1. Sean $p(x), q(x) \in K[x]$, con $q(x) \neq 0$. Entonces

$$mcd(p(x), q(x)) = mcd(q(x), p(x) \text{ mód } q(x)).$$

Para calcular ahora el máximo común divisor de dos polinomios procedemos de igual forma que a la hora de calcular el máximo común divisor de dos números enteros. Vamos realizando divisiones hasta obtener un resto nulo. En resto anterior es el máximo común divisor.

$$\begin{aligned}
p(x) &= q(x)\cdot c_1(x) + r_1(x)\\
q(x) &= r_1(x)\cdot c_2(x) + r_2(x)\\
r_1(x) &= r_2(x)\cdot c_3(x) + r_3(x)\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
r_{i-2}(x) &= r_{i-1}(x)\cdot c_i(x) + r_i(x)\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
r_{k-2}(x) &= r_{k-1}(x)\cdot c_k(x) + r_k(x)\\
r_{k-1}(x) &= r_k(x)\cdot c_{k+1}(x) + 0
\end{aligned}$$

Sin embargo, el polinomio $r_k(x)$ no tiene por qué ser mónico, luego el resultado final, $r_k(x)$, no sería el máximo común divisor de $p(x)$ y $q(x)$. Necesitamos multiplicar por el inverso del coeficiente líder para obtener el máximo común divisor.

Ejemplo:

Vamos a calcular en $\mathbb{Q}[x]$ el máximo común divisor de x^3-x+3 y x^3+x^2+1.

$$\begin{array}{rcrcl}
x^3-x+3 &=& (x^3+x^2+1)\ 1 & + & (-x^2-x+2)\\
x^3+x^2+1 &=& (-x^2-x+2)\ (-x) & + & 2x+1\\
-x^2-x+2 &=& (2x+1)\ \left(\frac{-1}{2}x-\frac{1}{4}\right) & + & \frac{9}{4}\\
2x+1 &=& \frac{9}{4}\ \left(\frac{8}{9}x+\frac{4}{9}\right) & + & 0
\end{array}$$

Luego un máximo común divisor de x^3-x+3 y x^3+x^2+1 es $\frac{9}{4}$. Multiplicamos por $\frac{4}{9}$ y obtenemos que $mcd(x^3-x+3, x^3+x^2+1)=1$.

3. Identidad de Bezout

El teorema de Bezout se tiene también en el caso de los polinomios.

Teorema 5.2. Sean $p(x), q(x) \in K[x]$, y sea $d(x) = mcd(p(x), q(x))$. Entonces existen $u(x), v(x) \in K[x]$ tales que $d(x) = p(x)\cdot u(x) + q(x)\cdot v(x)$

La forma de calcular los coeficientes $u(x)$ y $v(x)$ es análoga a la que se tiene para los números enteros.

Ejemplo:

1. Vamos a expresar $mcd(x^3-x+3, x^3+x^2+1)$ en función de los polinomios x^3-x+3 y x^3+x^2+1.

$p(x)$	$q(x)$	$r(x)$	$c(x)$	$u(x)$	$v(x)$
				1	0
				0	1
x^3-x+3	x^3+x^2+1	$-x^2-x+2$	1	1	-1
x^3+x^2+1	$-x^2-x+2$	$2x-1$	$-x$	x	$-x+1$
$-x^2-x+2$	$2x+1$	$\frac{9}{4}$	$\frac{-1}{2}x-\frac{1}{4}$	$\frac{1}{2}x^2+\frac{1}{4}x+1$	$\frac{-1}{2}x^2+\frac{1}{4}x-\frac{3}{4}$
$2x+1$	$\frac{9}{4}$	0			
	1			$\frac{2}{9}x^2+\frac{1}{9}x+\frac{4}{9}$	$\frac{-2}{9}x^2+\frac{1}{9}x-\frac{3}{9}$

Las cuatro primeras columnas es claro como se obtienen a partir del ejemplo anterior. En cuanto a las dos últimas, se han obtenido como sigue:

$$\begin{array}{ll} 1 = 1 - 1\cdot 0 & -1 = 0 - 1\cdot 1 \\ x = 0 - (-x)\cdot 1 & -x+1 = 1-(-x)\cdot(-1) \\ \frac{1}{2}x^2+\frac{1}{4}x+1 = 1-\left(\frac{-1}{2}x-\frac{1}{4}\right)\cdot x & \frac{-1}{2}x^2+\frac{3}{4}x-\frac{5}{4} = -1-\left(\frac{-1}{2}x-\frac{1}{4}\right)\cdot(-x+1) \end{array}$$

Nótese que se verifica que

$$1 = (x^3-x+3)\left(\frac{2}{9}x^2+\frac{1}{9}x+\frac{4}{9}\right)+(x^3+x^2+1)\left(\frac{-2}{9}x^2+\frac{1}{9}x-\frac{3}{9}\right)$$

2. Sean $p(x)=x^5+2x^4+x^2+2x+2,\ q(x)=x^5+2x^3+x^2+x+1\in\mathbb{Z}_3[x]$. Vamos a calcular su máximo común divisor y a exprearlo en función de $p(x)$ y $q(x)$.

$p(x)$	$q(x)$	$r(x)$	$c(x)$	$u(x)$	$v(x)$
				1	0
				0	1
$x^5+2x^4+x^2+2x+2$	$x^5+2x^3+x^2+x+1$	$2x^4+x^3+x+1$	1	1	2
$x^5+2x^3+x^2+x+1$	$2x^4+x^3+x+1$	$2x^2+2$	$2x+2$	$x+1$	$2x$
$2x^4+x^3+x+1$	$2x^2+2$	0			
	x^2+1			$2x+2$	x

Luego $\mathrm{mcd}(x^5+2x^4+x^2+2x+2, x^5+2x^3+x^2+x+1)=x^2+1$ y

$$x^2+1=(x^5+2x^4+x^2+2x+2)(2x+2)+(x^5+2x^3+x^2+x+1)(x)$$

3. En $\mathbb{Z}[x]$ se tiene que $\mathrm{mcd}(x,2)=1$. Sin embargo no existen $u(x),v(x)\in\mathbb{Z}[x]$ tales que $x\cdot u(x)+2\cdot v(x)=1$.

A partir de todo lo visto, es fácil probar que dados dos polinomios $p(x), q(x) \in K[x]$, existen $u(x), v(x) \in K[x]$ tales que

$$1 = p(x) \cdot u(x) + q(x) \cdot v(x)$$

si, y sólo si, $mcd(p(x), q(x)) = 1$.
También es fácil ver que si $p(x), q(x)$ son dos polinomios mónicos con coeficientes en K entonces $mcm(p(x), q(x)) = \frac{p(x)\cdot q(x)}{mcd(p(x),q(x))}$.

4. Ecuaciones diofánticas

Proposición 5.3. Sean $a(x), b(x), c(x) \in K[x]$. Entonces la ecuación $a(x)u(x) + b(x)v(x) = c(x)$ tiene solución si, y sólo si, $mcd(a(x), b(x))|c(x)$.
Si $u_0(x)$, $v_0(x)$ es una solución, y $d(x) = mcd(a(x), b(x))$, entonces todas las soluciones son de la forma:

$$\begin{array}{l} u(x) = u_0(x) + p(x)\frac{b(x)}{d(x)} \\ v(x) = v_0(x) - p(x)\frac{a(x)}{d(x)} \end{array} \qquad p(x) \in K[x]$$

Ejemplo:
Vamos a hallar todas las parejas de polinomio $u(x), v(x) \in \mathbb{Z}_3[x]$ que satisfacen la ecuación

$$(x^5 + 2x^3 + 2) \cdot u(x) + (x^5 + 2x^4 + 2x^3 + 1) \cdot v(x) = x^4 + 2x^2 + 2x + 2$$

Para esto, vemos en primer lugar si existe alguno. Esto ocurre si, y sólo si, $x^4 + 2x^2 + 2x + 2$ es múltiplo de $mcd(x^5 + 2x^3 + 2, x^5 + 2x^4 + 2x^3 + 1)$.

$a(x)$	$b(x)$	$r(x)$	$c(x)$
$x^5 + 2x^3 + 2$	$x^5 + 2x^4 + 2x^3 + 1$	$x^4 + 1$	1
$x^5 + 2x^4 + 2x^3 + 1$	$x^4 + 1$	$2x^3 + 2x + 2$	$x + 2$
$x^4 + 1$	$2x^3 + 2x + 2$	$2x^2 + 2x + 1$	$2x$
$2x^3 + 2x + 2$	$2x^2 + 2x + 1$	0	

luego $mcd(x^5+2x^3+2, x^5+2x^4+2x^3+1) = 2(2x^2+2x+1) = x^2+x+2$, y como $x^4+2x^2+2x+2 = (x^2+x+2)(x^2+2x+1)$ sabemos que podemos encontrar parejas de polinomio $u(x), v(x)$ que sean solución de la ecuación anterior.
Buscamos dos polinomios $u_0(x)$, $v_0(x)$ que sean solución

$a(x)$	$b(x)$	$r(x)$	$c(x)$	$u(x)$	$v(x)$
				1	0
				0	1
$x^5 + 2x^3 + 2$	$x^5 + 2x^4 + 2x^3 + 1$	$x^4 + 1$	1	1	2
$x^5 + 2x^4 + 2x^3 + 1$	$x^4 + 1$	$2x^3 + 2x + 2$	$x + 2$	$2x + 1$	$2x$
$x^4 + 1$	$2x^3 + 2x + 2$	$2x^2 + 2x + 1$	$2x$	$2x^2 + x + 1$	$x^2 + 2$
$2x^3 + 2x + 2$	$2x^2 + 2x + 1$	0			
	$x^2 + x + 2$			$x^2 + 2x + 2$	$2x^2 + 1$

Tomamos entonces

$$\begin{array}{rcrcl} u_0(x) & = & (x^2+2x+2)\cdot(x^2+2x+1) & = & x^4+x^3+x^2+2 \\ v_0(x) & = & (2x^2+1)\cdot(x^2+2x+1) & = & 2x^4+x^3+x^2+2x+2 \end{array}$$

Puesto que
$(x^5+2x^3+2) \text{ div } (x^2+x+2) = x^3+2x^2+x+2$
$(x^5+2x^4+2x^3+1) \text{ div } (x^2+x+2) = x^3+x^2+2x+2$
tenemos que la solución general es

$$\begin{array}{rcrcl} u(x) & = & x^4+x^3+x^2+2 & + & (x^3+x^2+2x+2)\cdot p(x) \\ v(x) & = & 2x^4+x^3+x^2+2x+2 & + & 2(x^3+2x^2+x+2)\cdot p(x) \end{array} \qquad p(x)\in\mathbb{Z}_3[x]$$

5. Factorización de polinomios

Definición 5. Sea $p(x)\in K[x]$ un polinomio no constante. Se dice que $p(x)$ es irreducible si sus únicos divisores son los polinomios constantes y los polinomios de la forma $a\cdot p(x)$, donde $a\in K^*$.

Ejemplo:
El polinomio $x^3+2x+2\in\mathbb{Z}_3[x]$ es irreducible. Podemos ver que sus únicos divisores son $1, 2, x^3+2x+2, 2x^3+x+1$.
Sin embargo, el polinomio $x^3+2x+2\in\mathbb{Z}_5[x]$ es reducible. Son divisores suyos, por ejemplo, $x+4, x+2, x^2+3x+1, x^2+x+3$.
Observación: Si $p(x)$ es un polinomio de grado n, y es reducible, entonces tiene un divisor cuyo grado está comprendido entre 1 y $\frac{n}{2}$ y es mónico.
Ejemplo:

1. Todo polinomio de grado 1 es irreducible.
2. Para ver que $x^3+2x+2\in\mathbb{Z}_3[x]$ es irreducible, basta ver que, al tener grado 3, si es reducible debe tener un divisor mónico de grado menor o igual que $\frac{3}{2}$, es decir, un divisor mónico de grado 1. Los únicos polinomios con coeficientes en $\mathbb{Z}_3$ de grado 1 y mónicos son $x, x+1$ y $x+2$, y ninguno de ellos lo divide.
3. Un polinomio con coeficientes complejos es irreducible si, y sólo si, es de grado 1.
4. Los polinomios irreducibles con coeficientes reales son los de grado 1 y los de grado 2 que tienen discriminante negativo.

También se podía haber definido un polinomio irreducible como aquel polinomio $p(x)$ no constante que satisface la siguiente propiedad:

$$p(x)|(q_1(x)\cdot q_2(x)) \Longrightarrow \big(p(x)|q_1(x) \text{ ó } p(x)|q_2(x)\big)$$

Veamos a continuación el teorema de factorización:

Teorema 5.3. Sea K un cuerpo, y $p(x) \in K[x]$ no constante. Entonces $p(x)$ se expresa de forma única (salvo el orden) como

$$p(x) = a \cdot p_1(x) \cdot p_2(x) \cdots p_k(x)$$

donde $a \in K$ y $p_i(x)$ es un polinomio mónico e irreducible.

El polinomio x^2+1 con coeficientes en $\mathbb{Z}_8$ se puede factorizar como $x^2+1 = (x+1)\cdot(x+7) = (x+3)\cdot(x+5)$. Al no ser $\mathbb{Z}_8$ un cuerpo, este ejemplo no contradice el teorema que acabamos de enunciar.

Vamos a ver ahora algunos resultados que nos van a facilitar la factorización de algunos polinomios con coeficientes en un cuerpo, y luego nos centraremos en la factorización de polinomios con coeficientes en un cuerpo finito.

Definición 6. Sea $p(x) \in K[x]$ y $a \in K$. Se dice que a es una raíz de $p(x)$ si $p(a) = 0$.

Ejemplo:
El polinomio $p(x) = x^5 + x^4 + x^3 + 2x^2 + 1 \in \mathbb{Z}_3[x]$ tiene a $x = 1$ por raíz, pues $p(1) = 1 + 1 + 1 + 2 + 1 = 0$. Sin embargo, 0 no es raíz pues $p(0) = 1$ y 2 tampoco es raíz pues $p(2) = 2^5 + 2^4 + 2^3 + 2 \cdot 2^2 + 1 = 2 + 1 + 2 + 2 + 1 = 2$.

El siguiente resultado es un conocido teorema referente a la división por el polinomio $x - a$.

Teorema 5.4 (Teorema del resto). Sea $p(x) \in K[x]$ y $a \in K$. Entonces el resto de dividir $p(x)$ entre $x - a$ es el resultado de evaluar $p(x)$ en el punto a. Dicho de otra forma

$$p(x) \text{ mód } x - a = p(a)$$

Demostración: Si dividimos $p(x)$ entre $x - a$ nos da un polinomio de grado menor que 1, luego debe ser un polinomio constante. Se tiene entonces que $p(x) = c(x) \cdot (x - a) + r$. Evaluando en a nos queda que $p(a) = c(a) \cdot (a - a) + r$, es decir, $r = p(a)$. ■

Corolario 5.2 (Teorema del factor). Sea $p(x) \in A[x]$ y $a \in A$. Entonces a es raíz de $p(x)$ si, y sólo si, $(x - a) | p(x)$.

En la siguiente proposición veremos una forma rápida de calcular el cociente y el resto de la división de un polinomio entre $x - a$.

Proposición 5.4. Sea $p(x) \in K[x]$, $a \in K$. Supongamos que $p(x) = a_n x^n + \cdots + a_1 x + a_0$ y que $p(x) = (b_{n-1}x^{n-1} + \cdots + b_1 x + b_0)(x - a) + r$. Entonces:

$b_{n-1} = a_n$

$b_{i-1} = a_i + b_i a$ para $i = 0, 1, \cdots, n-1$

$$r = a_0 + b_0 a$$

La demostración se deja como ejercicio.

Esta proposición proporciona el conocido método de Ruffini (algoritmo de Horner) para dividir un polinomio entre $x - a$.
Para esto se disponen los datos conocidos como sigue:

	a_n	a_{n-1}	$\cdots$	a_{i+1}	a_i	$\cdots$	a_1	a_0
a								
	b_{n-1}	b_{n-2}	$\cdots$	b_i	b_{i-1}	$\cdots$	b_0	r

Para calcular los coeficientes b_i se procede como sigue:
Se comienza por $b_{n-1} = a_n$
Supuesto calculado b_i se calcula b_{i-1} como $b_{i-1} = a_i + b_i a$.
Por último, hallado b_0 se calcula r como $r = a_0 + b_0 a$.
Para ordenar los cálculos se coloca el valor $b_i a$ justo debajo del valor de a_i, y se efectúa la suma, obteniéndose así el valor de b_{i-1}.

	a_n	a_{n-1}	$\cdots$	a_{i+1}	a_i	$\cdots$	a_1	a_0
a					$b_i a$			
	$b_{n-1} = a_n$	b_{n-2}	$\cdots$	b_i	$b_{i-1} = a_i + b_i a$	$\cdots$	b_0	r

Ejemplo:
Vamos a hallar el cociente y el resto de la división de $x^5 + x^4 + x^3 + 2x^2 + 1$ entre $x - 2$ en $\mathbb{Q}[x]$. Para ello procedemos a completar la tabla

	1	1	1	2	0	1
2						

Rellenando de izquierda a derecha.

	1	1		1		2		0		1
2		$2 = 1 \cdot 2$		$6 = 3 \cdot 2$		$14 = 7 \cdot 2$		$32 = 16 \cdot 2$		$64 = 32 \cdot 2$
	1	$3 = 1 + 2$		$7 = 1 + 6$		$16 = 2 + 14$		$32 = 0 + 32$		$65 = 1 + 64$

La tabla quedaría así

	1	1	1	2	0	1
2		2	6	14	32	64
	1	3	7	16	32	65

Nótese que $x^5 + x^4 + x^3 + 2x^2 + 1 = (x^4 + 3x^3 + 7x^2 + 16x + 32)(x - 2) + 65$, y que $p(2) = 65$.
Vamos a dividir ahora $x^5 + x^4 + x^3 + 2x^2 + 1$ entre $x + 1$ en $\mathbb{Z}_3[x]$. Puesto que $x + 1 = x - 2$, se tiene que

$$\begin{array}{c|cccccc} & 1 & 1 & 1 & 2 & 0 & 1 \\ 2 & & 2 & 0 & 2 & 2 & 1 \\ \hline & 1 & 0 & 1 & 1 & 2 & 2 \end{array}$$

es decir, el cociente es $x^4 + x^2 + x + 2$ y el resto es 2.

Definición 7. Sea $p(x) \in K[x]$, y $a \in K$. Se dice que a es una raíz de multiplicidad m si $(x-a)^m | p(x)$ y $(x-a)^{m+1} \not| p(x)$.

Nótese que decir que a es una raíz de multiplicidad m es decir que $p(x) = (x-a)^m c(x)$ con $c(a) \neq 0$.
A las raíces de multiplicidad 1 se les llama raíces simples; a las de multiplicidad 2, raíces dobles, a las de multiplicidad 3, raíces triples, y así sucesivamente.
En ocasiones, si a no es una raíz se dice que es una raíz de multiplicidad 0.

Ejemplo:
El polinomio $x^5 + x^3 + x^2 + 1 \in \mathbb{Z}_2[x]$ tiene a $x = 1$ como raíz triple, pues $x^5 + x^3 + x^2 + 1 = (x+1)^3(x^2 + x + 1)$, y $x^2 + x + 1$ no tiene a 1 como raíz.

$$\begin{array}{c|cccccc} & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & & 1 & 1 & 0 & 1 & 1 \\ \hline & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & & 1 & 0 & 0 & 1 & \\ \hline & 1 & 0 & 0 & 1 & 0 & \\ 1 & & 1 & 1 & 1 & & \\ \hline & 1 & 1 & 1 & 0 & & \\ 1 & & 1 & 0 & & & \\ \hline & 1 & 0 & 1 & & & \end{array}$$

Aquí vemos las sucesivas divisiones por $x + 1$. Se aprecia como las tres primeras son exactas, mientras que la cuarta da resto 1.
Un resultado que está implícitamente dicho en lo anterior, y que es muy útil en la práctica es:

Proposición 5.5. Sea $p(x) \in K[x]$, $gr(p(x)) = 2, 3$. Entonces $p(x)$ es irreducible si, y sólo si, $p(x)$ no tiene raíces.

Si el polinomio es de grado mayor o igual que 4 entonces el que no tenga raíces no nos permite afirmar que el polinomio sea irreducible.

Ejemplo:

1. El polinomio $p(x) = x^3 + 2x + 2 \in \mathbb{Z}_3[x]$ es irreducible. Al ser de grado 3 basta ver que no tiene raíces. Evaluamos en los tres elementos de $\mathbb{Z}_3$ y vemos que $p(0) = 2$, $p(1) = 2$ y $p(2) = 2$.

2. El polinomio $p(x) = x^4 + x^3 + x + 2 \in \mathbb{Z}_3[x]$ no tiene raíces ($p(0) = 2$, $p(1) = 2$ y $p(2) = 1$). Sin embargo no es irreducible, pues $p(x) = (x^2 + 1)(x^2 + x + 2)$.

Factorización de polinomios en $\mathbb{Z}_p[x]$

Aunque existen algoritmos para factorizar polinomios en $\mathbb{Z}_p[x]$ (algoritmo de Berlekamp), aquí emplearemos el método de ensayo y error.

Supongamos que tenemos un polinomio $q(x) \in \mathbb{Z}_p[x]$ de grado n. Si el polinomio es reducible, entonces tiene un factor irreducible de grado menor o igual que $\frac{n}{2}$.

Comprobamos en primer lugar si tiene o no divisores de grado 1, es decir, comprobamos si tiene raíces. A continuación comprobamos si tiene divisores irreducibles de grado 2, y así sucesivamente.

Ejemplo:

1. Sea $q(x) = x^3 + x + 1 \in \mathbb{Z}_2[x]$. Al ser de grado 3 únicamente hay que comprobar si tiene o no raíces. Puesto que $q(0) = q(1) = 1$ podemos deducir que el polinomio es irreducible. De la misma forma se comprueba que $x^3 + x^2 + 1$ es irreducible.

2. Sea ahora $q(x) = x^5 + x^4 + 1 \in \mathbb{Z}_2[x]$. En este caso $q(0) = q(1) = 1$, luego no tiene ningún divisor de grado 1.

   Probamos a dividir por $x^2 + x + 1$, que es irreducible de grado 2, y nos queda que $x^5 + x^4 + 1 = (x^2 + x + 1)(x^3 + x + 1)$. Los dos polinomios que aparecen son irreducibles.

3. Sea $q(x) = x^7 + x^4 + x^3 + x + 1 \in \mathbb{Z}_2[x]$. Entonces:

   Evaluamos en $x = 0$ y $x = 1$. En ambos casos nos sale 1, luego $q(x)$ no tiene divisores de grado 1.

   Dividimos por $x^2 + x + 1$, y nos queda $q(x) = (x^2 + x + 1)(x^5 + x^4 + x + 1) + x$. Por tanto no tiene divisores de grado 2.

   Dividimos por $x^3 + x + 1$ y $x^3 + x^2 + 1$. En el primer caso nos queda $q(x) = (x^3 + x + 1)(x^4 + x^2) + (x^2 + x + 1)$ y en el segundo $q(x) = (x^3 + x^2 + 1)(x^4 + x^3 + x^2 + x + 1)$.

   Puesto que $x^4 + x^3 + x^2 + x + 1$ no tiene divisores de grado 1 y grado 2 (ya que de tenerlos serían también divisores de $q(x)$) deducimos que $x^4 + x^3 + x^2 + x + 1$ es irreducible.

   La factorización de $q(x)$ como producto de irreducibles es

$$x^7 + x^4 + x^3 + x + 1 = (x^3 + x^2 + 1)(x^4 + x^3 + x^2 + x + 1).$$

Como vemos, para factorizar un polinomio en $\mathbb{Z}_p[x]$ es conveniente conocer los polinomios irreducibles mónicos de grado bajo, pues son por los que hemos de efectuar las divisiones. A continuación calcularemos algunos de estos irreducibles.

1. Polinomios irreducibles en $\mathbb{Z}_2[x]$

   - Grado 1. Aquí, los irreducibles son todos, es decir,
   $$x \qquad x+1.$$
   - Grado 2. Los no irreducibles son x^2, $x(x+1) = x^2+x$ y $(x+1)(x+1) = x^2+1$. El único que queda es
   $$x^2+x+1.$$
   - Grado 3. También aquí los únicos que hay son los que no tienen raíces. Estos son:
   $$x^3+x+1 \qquad x^3+x^2+1.$$
   - Grado 4. Aquí hemos de eliminar todos los que tengan raíces y $(x^2+x+1)^2 = x^4+x^2+1$. Nos quedan entonces tres polinomios, que son:
   $$x^4+x+1 \qquad x^4+x^3+1 \qquad x^4+x^3+x^2+x+1.$$
   - Grado 5. Los reducibles son los que tienen raíces y los dos que toman una factorización de la forma (grado 2) $\cdot$ (grado 3). Estos dos son $(x^2+x+1)(x^3+x+1) = x^5+x^4+1$ y $(x^2+x+1)(x^3+x^2+1) = x^5+x+1$.
   Nos quedan entonces 6 polinomios que son:
   $$x^5+x^2+1 \qquad x^5+x^3+1 \qquad x^5+x^4+x^3+x^2+1 \qquad x^5+x^4+x^3+x+1$$
   $$x^5+x^4+x^2+x+1 \qquad x^5+x^3+x^2+x+1.$$

2. Polinomios mónicos irreducibles en $\mathbb{Z}_3[x]$.

   - Grado 1. Al igual que antes, todos son irreducibles. Tenemos por tanto
   $$x \qquad x+1 \qquad x+2.$$
   - Grado 2. Son aquellos que no tiene raíces. Hay un total de 3, que son:
   $$x^2+1 \qquad x^2+x+2 \qquad x^2+2x+2.$$
   - Grado 3. Son también los que no tienen raíces. En este caso hay 8.
   $$x^3+2x+1 \qquad x^3+2x+2 \qquad x^3+x^2+2 \qquad x^3+2x^2+1$$
   $$x^3+x^2+x+2 \qquad x^3+x^2+2x+1 \qquad x^3+2x^2+x+1 \qquad x^3+2x^2+2x+2.$$

- De grado 4 hay 18 polinomios irreducibles.

3. Polinomios mónicos irreducibles en $\mathbb{Z}_5[x]$.
   - Grado 1. Tenemos 5 irreducibles:
   $$x \qquad x+1 \qquad x+2 \qquad x+3 \qquad x+4.$$
   - Grado 2. Los que no tienen raíces son 10.
   $$x^2+2 \qquad x^2+3 \qquad x^2+x+1 \qquad x^2+x+2 \qquad x^2+2x+3$$
   $$x^2+2x+4 \qquad x^2+3x+3 \qquad x^2+3x+4 \qquad x^2+4x+1 \qquad x^2+4x+2.$$
   - Para grados mayores el número de polinomios es muy grande. Así, de grado 3 la lista tendría 40 polinomios, mientras que la de grado 4 sería de 150.

4. Polinomios mónicos irreducibles en $\mathbb{Z}_7[x]$.
   - Grado 1. Como siempre aquí son todos irreducibles.
   $$x \qquad x+1 \qquad x+2 \qquad x+3 \qquad x+4 \qquad x+5 \qquad x+6.$$
   - Grado 2. Aquí la lista es ya muy grande. Tenemos un total de 21 polinomios.
   $$x^2+1 \quad x^2+2 \quad x^2+4 \quad x^2+x+3 \quad x^2+x+4 \quad x^2+x+6 \quad x^2+2x+2$$
   $$x^2+2x+3 \quad x^2+2x+5 \quad x^2+3x+1 \quad x^2+3x+5 \quad x^2+3x+6 \quad x^2+4x+1 \quad x^2+4x+5$$
   $$x^2+4x+6 \quad x^2+5x+2 \quad x^2+5x+3 \quad x^2+5x+5 \quad x^2+6x+3 \quad x^2+6x+4 \quad x^2+6x+6.$$
   - De grado 3 hay un total de 112 polinomios irreducibles.

En general, si denotamos por $I_{(p,d)}$ al número de polinomios irreducibles mónicos de grado d con coeficientes en $\mathbb{Z}_p$ se tiene que:

$$p^n = \sum_{d|n} d \cdot I_{(p,d)}$$

Por ejemplo, si $p = 2$ y $n = 2$ se tiene que $\sum_{d|2} d \cdot I_{(p,d)} = 1 \cdot I_{(2,1)} + 2 \cdot I_{(2,2)} = 1 \cdot 2 + 2 \cdot 1 = 4 = 2^2$, ya que hay dos irreducibles de grado 1 ($I_{(2,1)} = 2$) y un irreducible de grado 2 ($I_{(2,2)} = 1$). Como los divisores de 4 son $1, 2, 4$ se tiene que

$$16 = 2^4 = 1 \cdot I_{(2,1)} + 2 \cdot I_{(2,2)} + 4 \cdot I_{(2,4)} = 1 \cdot 2 + 2 \cdot 1 + 4 \cdot 3$$

Si queremos ver cuantos irreducibles hay de grado 8 podemos proceder como sigue:

$$256 = 2^8 = 1 \cdot I_{(2,1)} + 2 \cdot I_{(2,2)} + 4 \cdot I_{(2,4)} + 8 \cdot I_{(2,8)} = 16 + 8 \cdot I_{(2,8)} \Longrightarrow I_{(2,8)} = \frac{256-16}{8} = 30$$

Es decir, hay 30 irreducibles de grado 8 con coeficientes en $\mathbb{Z}_2$. Más adelante enumeraremos todos.

6. Congruencias.

Definición 8. Sea K un cuerpo y $a(x), b(x), m(x) \in K[x]$. Se dice que $a(x)$ es congruente con $b(x)$ módulo $m(x)$, y se escribe $a(x) \equiv b(x)(\text{mód } m(x))$ si $m(x)|(b(x) - a(x))$. Es decir:

$$a(x) \equiv b(x)(\text{mód } m(x)) \text{ si existe } c(x) \in K[x] \text{ tal que } b(x) - a(x) = c(x)m(x).$$

Nótese que la relación de congruencia módulo 0 es la relación de igualdad ($a(x) \equiv b(x)(\text{mód } 0)$ si, y sólo si, $a(x) = b(x)$), mientras que si $\lambda \in K^*$ entonces $a(x) \equiv b(x)(\text{mód } \lambda)$ cualesquiera que sean $a(x)$ y $b(x)$. Por tanto, nos centraremos en congruencias módulo $m(x)$ con $m(x)$ un polinomio de grado mayor o igual que 1.
Además, se tiene que $a(x) \equiv b(x)(\text{mód } m(x))$ si, y sólo si, $a(x) \equiv b(x)(\text{mód } \lambda \cdot m(x))$, donde $\lambda \in K^*$. Por tanto, al hablar de congruencias módulo $m(x)$ podemos suponer que $m(x)$ es un polinomio mónico.

Ejemplo:
Sea $m(x) = x^2 + 2 \in \mathbb{Z}_3[x]$. Entonces:

$x^4 + 2x^3 + x^2 + x + 2 \equiv 2x^4 + x^3 + 2x^2 + 2x(\text{mód } x^2 + 2)$
pues $(2x^4 + x^3 + 2x^2 + 2x) - (x^4 + 2x^3 + x^2 + x + 2) = (x^2 + 2)(x^2 + 2x + 2)$.

$x^4 + x^3 + 2x^2 + 1 \not\equiv x^3 + x + 2(\text{mód } x^2 + 2)$
ya que $(x^3 + x + 2) - (x^4 + x^3 + 2x^2 + 1) = 2x^2(x^2 + 2) + (x + 1)$.

Nos planteamos a continuación encontrar todos los polinomios $p(x) \in K[x]$ que verifican la relación

$$a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x))$$

con $a(x), b(x), m(x) \in K[x]$.
Un polinomio $q(x) \in K[x]$ para el que se verifique que $a(x) \cdot q(x) \equiv b(x)(\text{mód } m(x))$ es una solución de la congruencia.
Dos congruencias de la forma $a_1(x) \cdot p(x) \equiv b_1(x)(\text{mód } m_1(x))$ y $a_2(x) \cdot p(x) \equiv b_2(x)(\text{mód } m_2(x))$ son equivalentes si toda solución de la primera es solución de la segunda y viceversa.
La forma de resolver estas congruencias es análoga a la se emplea para resover congruencias en $\mathbb{Z}$.
Transformamos (si es posible) la congruencia $a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x))$ en otra equivalente de la forma $p(x) \equiv c(x)(\text{mód } n(x))$, cuyas soluciones son

$$p(x) = c(x) + q(x) \cdot n(x) : \quad q(x) \in K[x]$$

Los resultados necesarios para resolver estas congruencias son:

1. Si $a_1(x) \equiv a_2(x)(\text{mód } m(x))$ y $b_1(x) \equiv b_2(x)(\text{mód } m(x))$ entonces las congruencias $a_1(x) \cdot p(x) \equiv b_1(x)(\text{mód } m(x))$ y $a_2(x) \cdot p(x) \equiv b_2(x)(\text{mód } m(x))$ son equivalentes.

2. Si $d(x)$ es un divisor común de $a(x)$, $b(x)$ y $m(x)$, las congruencias

$$a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x)) \qquad \frac{a(x)}{d(x)}p(x) \equiv \frac{b(x)}{d(x)}(\text{mód } \frac{m(x)}{d(x)})$$

   son equivalentes.

3. Si $\text{mcd}(m(x), c(x)) = 1$ entonces las congruencias

$$a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x)) \qquad c(x) \cdot a(x) \cdot p(x) \equiv c(x) \cdot b(x)(\text{mód } m(x))$$

   son equivalentes.

Proposición 5.6. Sea K un cuerpo, y $a(x), b(x), m(x) \in K[x]$ tales que $\text{gr}(m(x)) \geq 1$. Entonces

$$a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x))$$

tiene solución si, y sólo si, $\text{mcd}(a(x), m(x))|b(x)$.

Para resolver congruencias de la forma $a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x))$ podemos proceder como sigue:

- Reducimos $a(x)$ y $b(x)$ módulo $m(x)$.
- Se comprueba si $\text{mcd}(a(x), m(x))|b(x)$. Si la respuesta es negativa, entonces la congruencia no tiene solución. Si la respuesta es afirmativa, podemos dividir toda la congruencia por $\text{mcd}(a(x), m(x))$.

  Hemos transformado la congruencia en una de la forma $a(x) \cdot p(x) \equiv b(x)(\text{mód } m(x))$, pero ahora se tiene que $\text{mcd}(a(x), m(x)) = 1$.
- Calculamos $u(x) \in K[x]$ tal que $a(x) \cdot u(x) \equiv 1(\text{mód } m(x))$.

  Este elemento existe, pues sabemos que la ecuación diofántica

$$1 = a(x) \cdot u(x) + m(x) \cdot v(x)$$

  tiene solución. Si $u_0(x), v_0(x)$ es una solución se tiene que

$$a(x) \cdot u_0(x) - 1 = (-v_0(x)) \cdot m(x) \Longrightarrow a(x) \cdot u_0(x) \equiv 1(\text{mód } m(x))$$

- Multiplicamos ambos miembros de la congruencia por $u(x)$. Obtenemos así una congruencia equivalente, y ésta adopta la forma $p(x) \equiv c(x)(\text{mód } m(x))$.

Con esto ya hemos resuelto la congruencia. Las soluciones son $p(x) = c(x) + q(x) \cdot m(x) : q(x) \in K[x]$.

Ejemplo:
Vamos a resolver en $\mathbb{Z}_{11}[x]$ la congruencia

$$(x^2 + 6x + 9)p(x) \equiv 3x^3 + 7x^2 + 9x + 2(\text{mód } x^3 + 5x^2 + 10x + 3)$$

Reducimos módulo $x^3 + 5x^2 + 10x + 2$.

$$(x^2 + 6x + 9) \cdot p(x) \equiv 3x^2 + x + 4(\text{mód } x^3 + 5x^2 + 10x + 3).$$

Hallamos el máximo común divisor de $x^2 + 6x + 9$ y $x^3 + 5x^2 + 10x + 3$.

$x^3 + 5x^2 + 10x + 3$	$x^2 + 6x + 9$	$7x + 1$	$x + 10$
$x^2 + 6x + 9$	$7x + 1$	3	$8x + 6$

Puesto que este máximo común divisor vale 1 resolvemos la ecuación

$$1 = m(x) \cdot v(x) + a(x) \cdot u(x)$$

				1	0
				0	1
$x^3 + 5x^2 + 10x + 3$	$x^2 + 6x + 9$	$7x + 1$	$x + 10$	1	$10x + 1$
$x^2 + 6x + 9$	$7x + 1$	3	$8x + 6$	$3x + 5$	$8x^2 + 9x + 6$
		1		$x + 9$	$10x^2 + 3x + 2$

Notemos que únicamente necesitamos el polinomio $10x^2+3x+2$, luego la penúltima columna no es necesario calcularla.
Multiplicamos por $10x^2 + 3x + 2$.

$$(10x^4 + 8x^3 + 6x + 7) \cdot p(x) \equiv 8x^4 + 8x^3 + 5x^2 + 3x + 8(\text{mód } x^3 + 5x^2 + 10x + 3).$$

Reducimos módulo $x^3 + 5x^2 + 10x + 3$.

$$p(x) \equiv 8x^2 + 2x + 5(\text{mód } x^3 + 5x^2 + 10x + 3).$$

Luego la solución es

$$p(x) = 8x^2 + 2x + 5 + c(x) \cdot (x^3 + 5x^2 + 10x + 3) : \quad c(x) \in \mathbb{Z}_{11}[x].$$

7. Teorema chino del resto y sistemas de congruencias

En lo referente a un sistema de congruencias se tiene también el teorema chino del resto.

Teorema 5.5. Sean $a_1(x), \cdots, a_k(x) \in K[x]$ y sean $m_1(x), \cdots, m_k(x) \in K[x]$ tales que

$\mathrm{mcd}(m_i(x), m_j(x)) = 1$. Entonces el sistema

$$\begin{aligned} p(x) &\equiv a_1(x)(\text{mód } m_1(x)) \\ p(x) &\equiv a_2(x)(\text{mód } m_2(x)) \\ &\cdots\cdots\cdots\cdots \\ p(x) &\equiv a_k(x)(\text{mód } m_k(x)) \end{aligned}$$

tiene solución. Además, si $a(x)$ es una solución, el sistema es equivalente a la congruencia

$$p(x) \equiv a(x)(\text{mód } M(x))$$

donde $M(x) = \prod_{i=1}^{k} m_i(x)$.

Sin embargo, a la hora de resolver sistemas de congruencias, procederemos a resolverlo progresivamente. Resolvemos la primera congruencia; introducimos esta solución en la segunda congruencia y la resolvemos; y así sucesivamente. De esta forma, no estamos sujetos a que se satisfagan las hipótesis del teorema chino. Veamos un ejemplo.

Ejemplo:
Vamos a resolver el sistema de congruencias en $\mathbb{Z}_5[x]$.

$$\begin{aligned} p(x) &\equiv x + 2(\text{mód } x^2 + 1) \\ (x+1) \cdot p(x) &\equiv x^2 + 1(\text{mód } x^3 + 2x^2 + 2) \\ x^2 \cdot p(x) &\equiv 3x + 2(\text{mód } x^2 + x + 1) \end{aligned}$$

Resolvemos la primera congruencia:

$$p(x) = x + 2 + (x^2 + 1) \cdot q_1(x).$$

Introducimos esta solución en la segunda congruencia.

$$\begin{aligned} &(x+1)(x+2+(x^2+1)\cdot q_1(x)) \equiv x^2+1(\text{mód } x^3+2x^2+2); \\ &x^2+3x+2+(x^3+x^2+x+1)\cdot q_1(x) \equiv x^2+1(\text{mód } \cdot)x^3+2x^2+2; \\ &(x^3+x^2+x+1)\cdot q_1(x) \equiv 2x+4(\text{mód } x^3+2x^2+2); \\ &(4x^2+x+4)\cdot q_1(x) \equiv 2x+4(\text{mód } x^3+2x^2+2). \end{aligned}$$

Calculamos $u(x)$ tal que $(x^2 + x + 4) \cdot u(x) \equiv 1(\text{mód } x^3 + 2x^2 + 2)$

				0
				1
x^3+2x^2+2	$4x^2+x+4$	$2x+4$	$4x+2$	$x+3$
$4x^2+x+4$	$2x+4$	3	$2x+4$	$3x^2+4$
		1		x^2+3

Multiplicamos entonces por x^2+3.

$$q_1(x) \equiv (x^2+3)\cdot(2x+4)(\text{mód } x^3+2x^2+2);$$
$$q_1(x) \equiv 2x^3+4x^2+x+2(\text{mód } x^3+2x^2+2);$$
$$q_1(x) \equiv x+3(\text{mód } x^3+2x^2+2).$$

Luego $q_1(x) = x+3+q_2(x)\cdot(x^3+2x^2+2)$ y por tanto

$$p(x) = x^3+3x^2+2x+q_2(x)\cdot(x^5+2x^4+x^3+4x^2+2)$$

Introducimos esta solución en la tercera congruencia, y operamos:

$$x^2\cdot(x^3+3x^2+2x)+x^2\cdot(x^5+2x^4+x^3+4x^2+2)\cdot q_2(x) \equiv 3x+2(\text{mód } x^2+x+1);$$
$$(x^7+2x^6+x^5+4x^4+2x^2)\cdot q_2(x) \equiv 4x^5+2x^4+3x^3+3x+2(\text{mód } x^2+x+1);$$
$$(2x+4)\cdot q_2(x) \equiv x+1(\text{mód } x^2+x+1).$$

Puesto que $(2x+4)\cdot(4x+1) \equiv 1(\text{mód } x^2+x+1)$ multiplicamos por $4x+1$.

$$q_2(x) \equiv (x+1)\cdot(4x+1)(\text{mód } x^2+x+1);$$
$$q_2(x) \equiv 4x^2+1(\text{mód } x^2+x+1);$$
$$q_2(x) \equiv x+2(\text{mód } x^2+x+1).$$

Por tanto, se tiene que $q_2(x) = x+2+q(x)(x^2+x+1)$. Introducimos este valor en lo que ya teníamos para $p(x)$ y nos queda:

$$p(x) = x^3+3x^2+2x+[x+2+(x^2+x+1)\cdot q(x)]\cdot(x^5+2x^4+x^3+4x^2+2),$$

es decir:

$$p(x) = x^6+4x^5+x^3+3x^2+2x+4+(x^7+3x^6+4x^5+2x^4+x^2+2x+2)\cdot q(x).$$

Un caso particularmente interesante es cuando queremos resolver un sistema de congruencias donde todos los módulos son polinomios mónicos de grado 1 (de la forma $x-a$). Para resolver este tipo de sistemas de congruencias es importante tener en cuenta que se verifica que

$$q(x) \equiv q(a)(\text{mód } x-a)$$

luego, para reducir un polinomio módulo $x-a$ basta con evaluar el polinomio en a.
Por otra parte, $q(a)^{-1}\cdot q(x) \equiv 1(\text{mód } x-a)$, donde $q(a)^{-1}$ representa el inverso de $q(a)$ en $K[x]$.
Por último, decir que encontrar un polinomio $p(x)$ que satisfaga la congruencia $p(x) \equiv b(\text{mód } x-a)$ es equivalente a encontrar un polinomio que verifique que $p(a) = b$.

Nos planteamos entonces el siguiente problema:

Dados $a_0, a_1, \cdots, a_m \in K$ todos distintos, y $b_0, b_1, \cdots, b_m \in K$, encontrar un polinomio $p(x) \in K[x]$ tal que $p(a_i) = b_i$.

Este problema se conoce como *problema de interpolación* y un polinomio solución se dice que es un polinomio interpolador.
Para resolverlo, planteamos el siguiente sistema de congruencias:

$$\begin{aligned} p(x) &\equiv b_0 (\text{mód } x - a_0) \\ p(x) &\equiv b_0 (\text{mód } x - a_0) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ p(x) &\equiv b_m (\text{mód } x - a_m) \end{aligned}$$

Cada una de las soluciones de este sistema será un polinomio interpolador.
Puesto que $mcd(x - a_i, x - a_j) = 1$ para $i \neq j$ deducimos, a partir del teorema chino, que este sistema tiene solución. Además, la solución es única módulo $\prod_{i=0}^{m}(x - a_i)$. Puesto que este polinomio tiene grado $m + 1$, deducimos que existe siempre un polinomio de grado menor o igual que m que interpola $m + 1$ datos.

Ejemplo:
Vamos a encontrar un polinomio en $\mathbb{Z}_7[x]$ que satisfaga que $p(1) = 2$, $p(2) = 5$, $p(4) = 6$ y $p(5) = 5$.
Para ello, planteamos el sistema de congruencias

$$\begin{aligned} p(x) &\equiv 2 (\text{mód } x + 6) \\ p(x) &\equiv 5 (\text{mód } x + 5) \\ p(x) &\equiv 6 (\text{mód } x + 3) \\ p(x) &\equiv 5 (\text{mód } x + 2) \end{aligned}$$

y procedemos a resolverlo como siempre:
Hallamos la solución de la primera congruencia

$$p(x) = 2 + (x + 6) \cdot q_1(x).$$

Introducimos esta solución en la segunda congruencia y operamos.

$$\begin{aligned} &2 + (x + 6) \cdot q_1(x) \equiv 5 (\text{mód } x + 5); \\ &(x + 6) \cdot q_1(x) \equiv 3 (\text{mód } x + 5); \\ &q_1(x) \equiv 3 (\text{mód } x + 5); \\ &q_1(x) = 3 + q_2(x) \cdot (x + 5). \end{aligned}$$

Luego resulta que $p(x) = 2 + (x + 6) \cdot [3 + q_2(x) \cdot (x + 5)] = 3x + 6 + (x + 6) \cdot (x + 5) \cdot q_2(x)$.
Continuamos introduciendo esta solución en la tercera congruencia.

$$3x + 6 + (x + 6) \cdot (x + 5) \cdot q_2(x) \equiv 6 (\text{mód } x + 3);$$

$$(x+6)\cdot(x+5)\cdot q_2(x) \equiv 4x(\text{mód } x+3);$$
$$6\cdot q_2(x) \equiv 2(\text{mód } x+3);$$
$$q_2(x) \equiv 5(\text{mód } x+3);$$
$$q_2(x) = 5 + q_3(x)\cdot(x+3).$$

Por tanto, $p(x) = 3x+6+(x+6)\cdot(x+5)\cdot[5+q_3(x)\cdot(x+3)] = 5x^2+2x+2+(x+6)\cdot(x+5)\cdot(x+3)\cdot q_3(x)$.

$$5x^2+2x+2+(x+6)\cdot(x+5)\cdot(x+3)\cdot q_3(x) \equiv 5(\text{mód } x+2);$$
$$(x+6)\cdot(x+5)\cdot(x+3)\cdot q_3(x) \equiv 2x^2+5x+3(\text{mód } x+2);$$
$$5\cdot q_3(x) \equiv 1(\text{mód } x+2);$$
$$q_3(x) \equiv 3(\text{mód } x+2);$$
$$q_3(x) = 3 + q(x)(x+2).$$

Nos queda entonces que $p(x) = 5x^2+2x+2+(x+6)\cdot(x+5)\cdot(x+3)\cdot[3+(x+2)\cdot q(x)]$, es decir,

$$p(x) = 3x^3+5x^2+2x+6+(x+6)\cdot(x+5)\cdot(x+3)\cdot(x+2)\cdot q(x),$$

luego una solución es $p(x) = 3x^3+5x^2+2x+6$.

Basándonos en esta idea podemos diseñar un algoritmo que calcule un polinomio que interpole unos datos dados. El polinomio interpolador obtenido se denomina *polinomio de interpolación de Newton*. Por tanto, denominaremos al algoritmo como NEWTON.

Algoritmo NEWTON$(m, a_0, b_0, a_1, b_1, \cdots, a_m, b_m)$

Entrada:

$m \in \mathbb{N}$

$a_0, b_0, a_1, b_1, \cdots, a_m, b_m \in K$

Salida: $p(x) \in K[x]$. $p(a_i) = b_i$ y $gr(p(x)) \leq n$

$p(x) := b_0$

$q(x) := x - a_0$

Desde $i = 1$ hasta m

$\quad p(x) := p(x) + q(a_i)^{-1}\cdot(b_i - p(a_i))\cdot q(x)$

$\quad q(x) := q(x)\cdot(x-a_i)$

Devuelve $p(x)$

Fin

Veamos como resover el ejemplo anterior haciendo uso de este algoritmo.

i	a_i	b_i	$q(a_i)$	$q(a_i)^{-1}$	$p(a_i)$	$b_i - p(a_i)$	$p(x)$	$q(x)$
							2	$x+6$
1	2	5	1	1	2	3	$3x+6$	x^2+4x+2
2	4	6	6	6	4	2	$5x^2+2x+2$	x^3+6
3	5	5	5	3	4	1	$3x^3+5x^2+2x+6$	x^4+2x^3+6x+5

Luego el polinomio interpolador es $p(x) = 3x^3 + 5x^2 + 2x + 6$. Todos los polinomios que satisfacen las condiciones dadas adoptan la forma:

$$p(x) = 3x^3 + 5x^2 + 2x + 6 + c(x) \cdot (x^4 + 2x^3 + 6x + 5) : \quad c(x) \in \mathbb{Z}_7[x]$$

8. Cocientes

Proposición 5.7. Sea $m(x) \in K[x]$. Entonces la relación de congruencia módulo $m(x)$ es una relación de equivalencia.

Para cada $m(x) \in K[x]$ vamos a denotar por $K[x]_{m(x)}$ al conjunto cociente de $K[x]$ por la relación de congruencia módulo $m(x)$. A la clase de equivalencia de un polinomio $a(x)$ la denotaremos inicialmente por $[a(x)]$.
Si $a, b, n \in \mathbb{Z}$ sabemos que $a \equiv b (\text{mód } n)$ si, y sólo si, $a \text{ mód } n = b \text{ mód } n$, es decir, $a \equiv b (\text{mód } n)$ si, y sólo si, a y b dan el mismo resto al dividir por n. Esto hacía que el número de elementos de $\mathbb{Z}_n$ fuera igual al número de posibles restos que se obtienen al dividir por n.
Pues ahora en $K[x]$ ocurre lo mismo: $a(x) \equiv b(x) (\text{mód } m(x))$ si, y sólo si, $a(x) \text{ mód } m(x) = b(x) \text{ mód } m(x)$, luego el conjunto $K[x]_{m(x)}$ está en biyección con los polinomios de $K[x]$ de grado menor que el de $m(x)$, pues éstos son los posibles restos de dividir un polinomio por $m(x)$.
Si $K = \mathbb{Z}_p$ y $gr(m(x)) = n$ entonces $K[x]_{m(x)} = p^n$.

Ejemplo:

1. Vamos a calcular los elementos del conjunto $\mathbb{Z}_2[x]_{(x^2+1)}$.

   Sea $p(x) \in \mathbb{Z}_2[x]$. Si dividimos $p(x)$ entre x^2+1, sólo tenemos cuatro posibles restos, que son 0, 1, x y $x+1$, ya que el resto es de grado menor que 2. Tenemos entonces que

   $$\mathbb{Z}_2[x]_{x^2+1} = \{[0], [1], [x], [x+1]\}.$$

   En la clase de equivalencia $[0]$ están todos los polinomios que dan resto cero al dividir por x^2+1, es decir, todos los múltiplos de x^2+1, por ejemplo, 0, x^2+1, x^3+x, x^4+1, etc.; en la clase $[1]$ están los polinomios que al dividir por x^2+1 dan resto 1, como por ejemplo, 1, x^2, x^3+x+1, x^4, etc.

2. El conjunto $\mathbb{Z}_2[x]_{x^2+x+1}$ tiene también cuatro elementos, que son $[0]$, $[1]$, $[x]$ y $[x+1]$. Sin embargo, aunque se representen igual que los de $\mathbb{Z}_2[x]_{x^2+1}$, los conjuntos $\mathbb{Z}_2[x]_{x^2+x+1}$ y $\mathbb{Z}_2[x]_{x^2+1}$ son distintos, pues en cada uno $[0]$, $[1]$, $[x]$ y $[x+1]$ representa cosas diferentes.

   Así, por ejemplo, en $\mathbb{Z}_2[x]_{x^2+x+1}$ se tiene que $[x^2+x]=[1]$, mientras que en $\mathbb{Z}_2[x]_{x^2+1}$, $[x^2+x]=[x+1]$.

3. El conjunto $\mathbb{Z}_2[x]_{x^3+x^2+x+1}$ tiene ocho elementos, mientras que $\mathbb{Z}_3[x]_{x^2+1}$ tiene nueve. Determínalos en ambos casos.

Lema 5.2. Sean $a(x), b(x), c(x), d(x), m(x) \in K[x]$. Entonces:

1. $$\left.\begin{array}{l} a(x) \equiv c(x) (\text{mód } m(x)) \\ b(x) \equiv d(x) (\text{mód } m(x)) \end{array}\right\} \Longrightarrow a(x)+b(x) \equiv c(x)+d(x) (\text{mód } m(x)).$$

2. $$\left.\begin{array}{l} a(x) \equiv c(x) (\text{mód } m(x)) \\ b(x) \equiv d(x) (\text{mód } m(x)) \end{array}\right\} \Longrightarrow a(x)b(x) \equiv c(x)d(x) (\text{mód } m(x)).$$

Y con este lema podemos ya definir las operaciones suma y producto

Definición 9. Sean $a(x), b(x) \in K[x]$ y $m(x) \in K[x]$ mónico y no constante. Se definen en $K[x]_{m(x)}$ las operaciones:

$$[a(x)]+[b(x)]=[a(x)+b(x)], \qquad [a(x)][b(x)]=[a(x)b(x)].$$

Como era de esperar, la definición hecha no depende de los representantes elegidos.

Ejemplo:
Supongamos que estamos trabajando en $\mathbb{Z}_3[x]_{x^2+1}$.
$[x+2]+[x+1]=[2x]$.
$[x+2][x+1]=[x^2+2]=[1]$.
Puesto que $[x+2]=[x^2+x]$ y $[x+1]=[2x^2+x]$ podíamos haber efectuado las operaciones anteriores
$[x^2+x]+[2x^2+x]=[3x^2+2x]=[2x]$.
$[x^2+x][2x^2+x]=[2x^4+x^2]=[1]$, ya que $2x^4+x^2=(x^2+1)(2x^2+2)+1$.
Y los resultados coinciden, como no podía ser de otra forma.

De ahora en adelante, si $a \in K \subseteq K[x]$, denotaremos por a a la clase de equivalencia $[a] \in K[x]_{m(x)}$, mientras que denotaremos por α (u otra letra griega, salvo que especifiquemos lo contrario) a la clase de equivalencia $[x]$.
Nótese que siguiendo esta notación, dado $a_k x^k+\cdots+a_1 x+a_0 \in K[x]$ el elemento $[a_k x^k+\cdots+a_1 x+a_0]$ se representa como $a_k\alpha^k+\cdots+a_1\alpha+a_0$. Dicho de otra forma, $[p(x)]$ se representa como $p(\alpha)$.

Nótese también que con esta notación se verifica que $m(\alpha) = 0$, pues $m(\alpha) = [m(x)] = [0]$. Además, esta condición es suficiente para realizar las operaciones en $K[x]_{m(x)}$

$$K[x]_{m(x)} = \{p(\alpha) : p(x) \in K[x]; m(\alpha) = 0\}.$$

Ejemplo:

1. En el conjunto $\mathbb{Z}_2[x]_{x^3+x+1}$ vamos a multiplicar $[x^2+x+1]$ y $[x^2+1]$. Podemos proceder de dos formas:

   a) Multiplicamos los dos polinomios:
   $[x^2 + x + 1] \cdot [x^2 + 1] = [x^4 + x^3 + x + 1]$.
   Dividimos x^4+x^3+x+1 entre x^3+x+1. $x^4+x^3+x+1 = (x^3+x+1)\cdot(x+1)+x^2+x$.
   Por tanto $[x^2 + x + 1] \cdot [x^2 + 1] = [x^2 + x]$.

   b) $(\alpha^2 + \alpha + 1) \cdot (\alpha^2 + 1) = \alpha^4 + \alpha^3 + \alpha + 1$.
   Puesto que $\alpha^3 + \alpha + 1 = 0$ deducimos que $\alpha^3 = \alpha + 1$, luego $\alpha^4 = \alpha^2 + \alpha$. Por tanto $(\alpha^2 + \alpha + 1) \cdot (\alpha^2 + 1) = \alpha^4 + \alpha^3 + \alpha + 1 = (\alpha^2 + \alpha) + (\alpha + 1) + \alpha + 1 = \alpha^2 + \alpha$.

   En los dos casos se obtiene el mismo resultado.

2. $\mathbb{Z}_2[x]_{x^2+1} = \{0, 1, \alpha, \alpha + 1 : \alpha^2 + 1 = 0\}$, o si preferimos:

$$\mathbb{Z}_2[x]_{x^2+1} = \{0, 1, \alpha, \alpha + 1 : \alpha^2 = 1\}.$$

Proposición 5.8. Sea $m(x) \in k[x]$ mónico y no constante. Las operaciones suma y producto en $K[x]_{m(x)}$ verifican las siguientes propiedades:

i) $p(\alpha) + (q(\alpha) + r(\alpha)) = (p(\alpha) + q(\alpha)) + r(\alpha)$

ii) $p(\alpha) + q(\alpha) = q(\alpha) + p(\alpha)$

iii) $p(\alpha) + 0 = p(\alpha)$

iv) Para cada $p(\alpha) \in K[x]_{m(x)}$ existe $q(\alpha) \in K[x]_{m(x)}$ tal que $p(\alpha) + q(\alpha) = 0$.

v) $p(\alpha) \cdot (q(\alpha) \cdot r(\alpha)) = (p(\alpha) \cdot q(\alpha)) \cdot r(\alpha)$

vi) $p(\alpha) \cdot q(\alpha) = q(\alpha) \cdot p(\alpha)$

vii) $p(\alpha) \cdot 1 = p(\alpha)$

viii) $p(\alpha) \cdot (q(\alpha) + r(\alpha)) = p(\alpha) \cdot q(\alpha) + p(\alpha) \cdot r(\alpha)$

Estas propiedades nos dicen que $K[x]_{m(x)}$ es un anillo conmutativo.

Ejemplo:

1. Consideramos el anillo $\mathbb{Z}_2[x]_{x^3+1}$. Vamos a escribir las tablas de sumar y multiplicar de dicho anillo. Antes de ello, enumeramos sus elementos

$$\mathbb{Z}_2[x]_{x^3+1} = \{0, 1, \alpha, \alpha+1, \alpha^2, \alpha^2+1, \alpha^2+\alpha, \alpha^2+\alpha+1\}$$

$+$	0	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
0	0	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
1	1	0	$\alpha+1$	α	α^2+1	α^2	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$
α	α	$\alpha+1$	0	1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	α^2	α^2+1
$\alpha+1$	$\alpha+1$	α	1	0	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	α^2+1	α^2
α^2	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	0	1	α	$\alpha+1$
α^2+1	α^2+1	α^2	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	1	0	$\alpha+1$	α
$\alpha^2+\alpha$	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	α^2	α^2+1	α	$\alpha+1$	0	1
$\alpha^2+\alpha+1$	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	α^2+1	α^2	$\alpha+1$	α	1	0

Para realizar la tabla del producto tenemos en cuenta que $\alpha^3+1=0$, es decir, $\alpha^3=1$.

$\cdot$	0	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
0	0	0	0	0	0	0	0	0
1	0	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
α	0	α	α^2	$\alpha^2+\alpha$	1	$\alpha+1$	α^2+1	$\alpha^2+\alpha+1$
$\alpha+1$	0	$\alpha+1$	$\alpha^2+\alpha$	α^2+1	α^2+1	$\alpha^2+\alpha$	$\alpha+1$	0
α^2	0	α^2	1	α^2+1	α	$\alpha^2+\alpha$	$\alpha+1$	$\alpha^2+\alpha+1$
α^2+1	0	α^2+1	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^2+\alpha$	$\alpha+1$	α^2+1	0
$\alpha^2+\alpha$	0	$\alpha^2+\alpha$	α^2+1	$\alpha+1$	$\alpha+1$	α^2+1	$\alpha^2+\alpha$	0
$\alpha^2+\alpha+1$	0	$\alpha^2+\alpha+1$	$\alpha^2+\alpha+1$	0	$\alpha^2+\alpha+1$	0	0	$\alpha^2+\alpha+1$

Donde algunas de las casillas se han completado como sigue:

$$\alpha \cdot \alpha^2 = \alpha^3 = 1$$

$$(\alpha^2+1)\cdot(\alpha^2+\alpha+1) = \alpha^4+\alpha^3+\alpha^2+\alpha^2+\alpha+1 = \alpha+1+\alpha^2+\alpha^2+\alpha+1 = 0$$

Y aquí se ha tenido en cuenta que $\alpha^4 = \alpha^3 \cdot \alpha = \alpha$.

$$(\alpha^2+1)\cdot(\alpha^2+1) = \alpha^4+2\alpha^2+1 = \alpha+1.$$

2. Vamos a dar ahora la tabla de multiplicar de $\mathbb{Z}_3[x]_{x^2+1}$. Los elementos son ahora

$$\mathbb{Z}_3[x]_{x^2+1} = \{0, 1, 2, \alpha, \alpha+1, \alpha+2, 2\alpha, 2\alpha+1, 2\alpha+2\}$$

$\cdot$	0	1	2	α	$\alpha+1$	$\alpha+2$	2α	$2\alpha+1$	$2\alpha+2$
0	0	0	0	0	0	0	0	0	0
1	0	1	2	α	$\alpha+1$	$\alpha+2$	2α	$2\alpha+1$	$2\alpha+2$
2	0	2	1	2α	$2\alpha+2$	$2\alpha+1$	α	$\alpha+2$	$\alpha+1$
α	0	α	2α	2	$\alpha+2$	$2\alpha+2$	1	$\alpha+1$	$2\alpha+1$
$\alpha+1$	0	$\alpha+1$	$2\alpha+2$	$\alpha+2$	2α	1	$2\alpha+1$	2	α
$\alpha+2$	0	$\alpha+2$	$2\alpha+1$	$2\alpha+2$	1	α	$\alpha+1$	2α	2
2α	0	2α	α	1	$2\alpha+1$	$\alpha+1$	2	$2\alpha+2$	$\alpha+2$
$2\alpha+1$	0	$2\alpha+1$	$\alpha+2$	$\alpha+1$	2	2α	$2\alpha+2$	α	1
$2\alpha+2$	0	$2\alpha+2$	$\alpha+1$	$2\alpha+1$	α	2	$\alpha+2$	1	2α

Definición 10. Sea A un anillo conmutativo.

- Se dice que $a \in A$ es una unidad si existe $b \in A$ tal que $a \cdot b = 1$. En tal caso, se dice que b es el inverso de a, y escribiremos $b = a^{-1}$.
- Se dice que $a \in A$ es un divisor de cero si existe $b \in A$, $b \neq 0$ tal que $a \cdot b = 0$.

Un anillo conmutativo en el que 0 es el único divisor de cero se llama *dominio de integridad*. Un anillo conmutativo en el que todo elemento no nulo es una unidad es lo que llamamos al principio un cuerpo.

Ejemplo:

1. En cualquier anillo, 1 es una unidad, pues $1 \cdot 1 = 1$, mientras que 0 es un divisor de cero, pues $0 \cdot 1 = 0$.
2. En $\mathbb{Z}_2[x]_{x^3+1}$ son divisores de cero:

   $$0 \qquad \alpha+1 \qquad \alpha^2+1 \qquad \alpha^2+\alpha \qquad \alpha^2+\alpha+1$$

   mientras que son unidades:

   $$1 \qquad \alpha \qquad \alpha^2$$

   como puede comprobarse a partir del ejemplo anterior.
3. En $\mathbb{Z}_3[x]_{x^2+1}$, el único divisor de cero es 0. Todos los demás elementos son unidades.
4. En $\mathbb{Z}$, las unidades son 1 y -1. El único divisor de cero es 0.
5. Todo cuerpo es un dominio de integridad. El recíproco no es cierto, pues $\mathbb{Z}$ es un dominio de integridad pero no es un cuerpo.

Proposición 5.9. Sea K un cuerpo, $m(x) \in K[x]$ no constante y $p(\alpha) \in K[x]_{m(x)}$. Entonces:

- $p(\alpha)$ es una unidad si, y sólo si, $mcd(p(x), m(x)) = 1$.

- $p(\alpha)$ es un divisor de cero si, y sólo si, $mcd(p(x), m(x)) \neq 1$.

Demostración: La primera parte es consecuencia de que la ecuación $1 = p(x)\cdot u(x)+m(x)\cdot v(x)$ tiene solución si, y sólo si, $mcd(p(x), m(x)) = 1$.
En cuanto a la segunda, si $p(\alpha)$ es un divisor de cero, entonces $p(\alpha)$ no es una unidad (¿por qué?), luego $mcd(p(x), m(x)) \neq 1$.
Recíprocamente, si $mcd(p(x), m(x)) \neq 1$, consideramos $q(x) = \frac{m(x)}{d(x)}$ donde $d(x) = mcd(p(x), m(x))$. Entonces $gr(q(x)) < gr(m(x))$, lo que implica que $q(\alpha) \neq 0$, y puesto que $p(x)\cdot q(x)$ es múltiplo de $m(x)$ ya que

$$p(x) \cdot q(x) = p(x)\frac{m(x)}{d(x)} = \frac{p(x)}{d(x)}m(x)$$

se verifica que $p(\alpha) \cdot q(\alpha) = 0$. ■

Ejemplo:
En $\mathbb{Z}_2[x]$ se verifica que $mcd(x^2 + 1, x^3 + 1) = x + 1$. Por tanto, $\alpha^2 + 1$ es un divisor de cero en $\mathbb{Z}_2[x]_{x^3+1}$. Además, para encontrar un elemento que al multiplicarlo por él nos de cero, calculamos $\frac{x^3+1}{x+1}$. Ese cociente vale x^2+x+1. Deducimos entonces que $(\alpha^2+1)\cdot(\alpha^2+\alpha+1) = 0$, como podemos ver en el ejemplo anterior.

A partir de la proposición anterior se deduce fácilmente que si $m(x)$ es un polinomio irreducible en $K[x]$, entonces $K[x]_{m(x)}$ es un cuerpo. Si $m(x)$ es un polinomio irreducible de grado n en $\mathbb{Z}_p[x]$ entonces $\mathbb{Z}_p[x]_{m(x)}$ es un cuerpo con p^n elementos.
Por otra parte, si K es un cuerpo con un número finito de elementos, entonces su característica es un número primo p (la característica de un anillo A se define como el menor número natural n tal que $1 + 1 + \overset{n \text{ veces}}{\cdots} + 1 = 0$, si dicho número existe). En tal caso se tiene que $\mathbb{Z}_p \subseteq K$, lo que implica que K es un $\mathbb{Z}_p$-espacio vectorial. Si $n = dim_{\mathbb{Z}_p}(K)$ entonces K tiene p^n elementos.
Es decir, por una parte hemos visto que el número de elementos de un cuerpo finito es una potencia de un primo. Por otra parte, hemos visto como, dado un número primo p y un número natural n podemos construir un cuerpo con p^n elementos. Basta encontrar un polinomio irreducible de grado n en $\mathbb{Z}_p[x]$. Hay un teorema que nos asegura la existencia de polinomios irreducibles de cualquier grado en $\mathbb{Z}_p[x]$.
La existencia de varios polinomios irreducibles de un mismo grado en $\mathbb{Z}_p[x]$ daría lugar, en principio, a distintos cuerpos con p^n elementos. Sin embargo, todos los cuerpos con el mismo cardinal son isomorfos, en el sentido que vamos a explicar a continuación.

Ejemplo:

1. Como $x^2 + x + 1 \in \mathbb{Z}_2[x]$, entonces $\mathbb{Z}_2[x]_{x^2+x+1}$ es un cuerpo. Escribamos su tabla de la suma y el producto:

$+$	0	1	α	$\alpha+1$
0	0	1	α	$\alpha+1$
1	1	0	$\alpha+1$	α
α	α	$\alpha+1$	0	1
$\alpha+1$	$\alpha+1$	α	1	0

$\cdot$	0	1	α	$\alpha+1$
0	0	0	0	0
1	0	1	α	$\alpha+1$
α	0	α	$\alpha+1$	1
$\alpha+1$	0	$\alpha+1$	1	α

Si nos fijamos en las tablas del ejemplo que dimos al principio, y sustituimos en ellas a por α y b por $\alpha+1$, obtenemos las tablas de este ejemplo.

2. Hemos visto que $\mathbb{Z}_3[x]_{x^2+1}$ es un cuerpo con nueve elementos, cuya tabla del producto calculamos en el ejemplo ??. Puesto que x^2+x+2 es también un polinomio irreducible en $\mathbb{Z}_3[x]$ tenemos que $\mathbb{Z}_3[x]_{x^2+x+2}$ es también un cuerpo con nueve elementos. Si llamamos β al elemento $[x]$, entonces la tabla del producto de este cuerpo es:

$(\mathbb{Z}_3[x]_{x^2+x+2},\cdot)$	0	1	2	β	$\beta+1$	$\beta+2$	2β	$2\beta+1$	$2\beta+2$
0	0	0	0	0	0	0	0	0	0
1	0	1	2	β	$\beta+1$	$\beta+2$	2β	$2\beta+1$	$2\beta+2$
2	0	2	1	2β	$2\beta+2$	$2\beta+1$	β	$\beta+2$	$\beta+1$
β	0	β	2β	$2\beta+1$	1	$\beta+1$	$\beta+2$	$2\beta+2$	2
$\beta+1$	0	$\beta+1$	$2\beta+2$	1	$\beta+2$	2β	2	β	$2\beta+1$
$\beta+2$	0	$\beta+2$	$2\beta+1$	$\beta+1$	2β	2	$2\beta+2$	1	β
2β	0	2β	β	$\beta+2$	2	$2\beta+2$	$2\beta+1$	$\beta+1$	1
$2\beta+1$	0	$2\beta+1$	$\beta+2$	$2\beta+2$	β	1	$\beta+1$	2	2β
$2\beta+2$	0	$2\beta+2$	$\beta+1$	2	$2\beta+1$	β	1	2β	$\beta+2$

donde se ha usado que $\beta^2=2\beta+1$, relación que se deduce de $\beta^2+\beta+2=0$ (es decir, $m(\beta)=0$).

Si ahora hacemos el cambio $\alpha=\beta+2$, es decir, $\beta=\alpha+1$, la tabla nos quedaría

$(\mathbb{Z}_3[x]_{x^2+x+2},\cdot)$	0	1	2	$\alpha+1$	$\alpha+2$	α	$2\alpha+2$	2α	$2\alpha+1$
0	0	0	0	0	0	0	0	0	0
1	0	1	2	$\alpha+1$	$\alpha+2$	α	$2\alpha+2$	2α	$2\alpha+1$
2	0	2	1	$2\alpha+2$	$2\alpha+1$	2α	$\alpha+1$	α	$\alpha+2$
$\alpha+1$	0	$\alpha+1$	$2\alpha+2$	2α	1	$\alpha+2$	α	$2\alpha+1$	2
$\alpha+2$	0	$\alpha+2$	$2\alpha+1$	1	α	$2\alpha+2$	2	$\alpha+1$	2α
α	0	α	2α	$\alpha+2$	$2\alpha+2$	2	$2\alpha+1$	1	$\alpha+1$
$2\alpha+2$	0	$2\alpha+2$	$\alpha+1$	α	2	$2\alpha+1$	2α	$\alpha+2$	1
2α	0	2α	α	$2\alpha+1$	$\alpha+1$	1	$\alpha+2$	2	$2\alpha+2$
$2\alpha+1$	0	$2\alpha+1$	$\alpha+2$	2	2α	$\alpha+1$	1	$2\alpha+2$	α

Si comparamos esta tabla con la que obtuvimos para $\mathbb{Z}_3[x]_{x^2+1}$ vemos que es exactamente la misma (salvo el orden de las filas y columnas). Vemos entonces que los cuerpos $\mathbb{Z}_3[x]_{x^2+1}$ y $\mathbb{Z}_3[x]_{x^2+x+2}$ son iguales, o más precisamente, son isomorfos.

De hecho, lo único que diferencia a los cuerpos $\mathbb{Z}_3[x]_{x^2+1}$ y $\mathbb{Z}_3[x]_{x^2+x+2}$ es, aparte del camino para obtenerlos, el nombre que se le ha dado a los elementos. Lo que en un cuerpo se llama α en el otro se llama $\beta+2$. Una vez hecha la correcta correspondencia entre los elementos de uno y del otro, se opera de igual forma en un caso y en el otro.

Nota: Dados dos cuerpos K y K', se dice que son isomorfos si existe una aplicación $f : K \to K'$ satisfaciendo:

a) f preserva la suma, es decir, $f(a+b) = f(a) + f(b)$.

b) f preserva el producto, es decir, $f(a \cdot b) = f(a) \cdot f(b)$.

c) f es biyectiva.

f es lo que se llama un *isomorfismo de cuerpos*.

En el caso de $K = \mathbb{Z}_3[x]_{x^2+x+2}$ y $K' = \mathbb{Z}_3[x]_{x^2+1}$, la aplicación $f : K \to K'$ dada por

$$0 \mapsto 0 \quad 1 \mapsto 1 \quad 2 \mapsto 2 \quad \beta \mapsto \alpha+1 \quad \beta+1 \mapsto \alpha+2$$

$$\beta+2 \mapsto \alpha \quad 2\beta \mapsto 2\alpha+2 \quad 2\beta+1 \mapsto 2\alpha \quad 2\beta+2 \mapsto 2\alpha+1$$

es un isomorfismo de cuerpos. Obviamente, este isomorfismo queda totalmente determinado por $\beta \mapsto \alpha+1$.

3. Nos situamos en el cuerpo de los números reales. Entonces el polinomio x^2+1 es irreducible, luego $\mathbb{R}[x]_{x^2+1}$ es un cuerpo. Si llamamos i al elemento $[x]$, entonces se tiene que los elementos de $\mathbb{R}[x]_{x^2+1}$ son de la forma $a+bi$, donde $a, b \in \mathbb{R}$. Además, $i^2+1 = 0$, es decir, $i^2 = -1$.

   Por tanto,

   $$\mathbb{R}[x]_{x^2+1} = \{a+bi : a, b \in \mathbb{R}; i^2 = -1\}$$

   luego el cuerpo obtenido resulta ser igual (o isomorfo) a $\mathbb{C}$.

Dado p es un número primo y n es un número natural no nulo, denotaremos como $\mathbb{F}_{p^n}$ al único cuerpo que existe con p^n elementos. Así, por ejemplo, $\mathbb{F}_4 = \mathbb{Z}_2[x]_{x^2+x+1}$ y $\mathbb{F}_9 = \mathbb{Z}_3[x]_{x^2+1}$. Obviamente, $\mathbb{F}_p = \mathbb{Z}_p$ para cualquier primo p.

. . . . . 5.2

El cuerpo $\mathbb{F}_{256}$

Acabamos de ver que para cada número primo p y cada número natural distinto de cero n hay un cuerpo con p^n elementos, que se denomina $\mathbb{F}_{p^n}$. Vamos a estudiar con detalle el caso $p = 2$ y $n = 8$, ya que el algoritmo de cifrado AES trabaja con los elementos de ese cuerpo.

Para construir $\mathbb{F}_{256}$ necesitamos un polinomio de grado 8, con coeficientes en $\mathbb{Z}_2$ y que sea irreducible. Hemos visto que hay 30 polinomios que satisfacen esas condiciones. Esos polinomios son:

$x^8+x^4+x^3+x+1$	$x^8+x^4+x^3+x^2+1$	$x^8+x^5+x^3+x+1$
$x^8+x^5+x^3+x^2+1$	$x^8+x^5+x^4+x^3+1$	$x^8+x^5+x^4+x^3+x^2+x+1$
$x^8+x^6+x^3+x^2+1$	$x^8+x^6+x^4+x^3+x^2+x+1$	$x^8+x^6+x^5+x+1$
$x^8+x^6+x^5+x^2+1$	$x^8+x^6+x^5+x^3+1$	$x^8+x^6+x^5+x^4+1$
$x^8+x^6+x^5+x^4+x^2+x+1$	$x^8+x^6+x^5+x^4+x^3+x+1$	$x^8+x^7+x^2+x+1$
$x^8+x^7+x^3+x+1$	$x^8+x^7+x^3+x^2+1$	$x^8+x^7+x^4+x^3+x^2+x+1$
$x^8+x^7+x^5+x+1$	$x^8+x^7+x^5+x^3+1$	$x^8+x^7+x^5+x^4+1$
$x^8+x^7+x^5+x^4+x^3+x^2+1$	$x^8+x^7+x^6+x+1$	$x^8+x^7+x^6+x^3+x^2+x+1$
$x^8+x^7+x^6+x^4+x^2+x+1$	$x^8+x^7+x^6+x^4+x^3+x^2+1$	$x^8+x^7+x^6+x^5+x^2+x+1$
$x^8+x^7+x^6+x^5+x^4+x+1$	$x^8+x^7+x^6+x^5+x^4+x^2+1$	$x^8+x^7+x^6+x^5+x^4+x^3+1$

Cualquiera de los 30 polinomios nos valdría para definir $\mathbf{F}_{256}$, y no añade ninguna seguridad en AES emplear uno u otro. Se tomó entonces el primero, es decir, $x^8+x^4+x^3+x+1$. Por tanto, vamos a trabajar en el cuerpo $\mathbb{Z}_{2_{x^8+x^4+x^3+x+1}}$.
Los elementos del cuerpo son ahora clases de equivalencia de polinomios de grado menor que 8. Por ejemplo, son elementos de $\mathbb{F}_{256}$:

$$[x^5+x^2+1] \quad [x^7+x] \quad 0 \quad [x^2] \quad [x^7+x^5+x^3+x]$$

Sin embargo, a la hora de representar cada elemento podemos hacerlo de otras formas:
Puesto que cada elemento se corresponde con un polinomio de grado menor o igual que 7, está determinado por los 8 coeficientes. Cada uno de estos coeficientes está en $\mathbb{Z}_2$, luego es 0 o es 1 (es decir un bit). Por tanto, cada elemento puede ser representado como una cadena de 8 bits (es decir, un byte). Así, los 5 elementos anteriores se podrían representar como:

$$00100101 \qquad 10000010 \qquad 00000000 \qquad 00000100 \qquad 10101010$$

A su vez cada byte podemos representarlo en notación hexadecimal. Cada 4 bits se corresponden con un dígito hexadecimal, luego un byte lo representamos mediante dos dígitos hexadecimales. Los cinco elementos anteriores nos quedarían entonces:

$$25 \qquad 82 \qquad 00 \qquad 04 \qquad AA$$

Por último, podemos emplear la notación decimal. Por tanto cada elemento puede ser visto con un número comprendido entre 0 y 255.

$$37 \qquad 130 \qquad 0 \qquad 4 \qquad 170$$

Nos quedarían entonces cuatro representaciones: polinomial, binaria, hexadecimal y decimal

Polinomial	Binaria	Hexadecimal	Decimal
$[x^5+x^2+x+1]$	00100111	27	39
$[x^7+x^6+x^5+x^3+x+1]$	11101011	EB	235
$[1]$	00000001	01	1
$[x+1]$	00000011	03	3
$[x^3+x^2+x+1]$	00001111	0F	15

En $\mathbb{F}_{256}$ sabemos que tenemos dos operaciones, la suma y el producto. Para la suma, basta hacer la suma en $\mathbb{Z}_2$. En notación binaria, la suma se reduce a un XOR, es decir, a la suma bit a bit.

El opuesto (para la suma) de cualquier elemento es él mismo. Por tanto, sumar y restar es lo mismo.

Así, en notación binaria se tiene que $11010110+01110101 = 10100011$. Esta suma, en notación hexadecimal sería $D6+75 = A3$, mientras que en decimal tendríamos $214+117 = 163$

El producto es algo más complicado. En principio, hay que realizar el producto en notación polinomial, y posteriormente dividir por $x^8+x^4+x^3+x+1$. Por ejemplo, vamos a multiplicar $[x^7+x^6+x^4+x^2+x]$ por $[x^6+x^5+x^4+x^2+1]$.

Multiplicamos los dos polinomios:

$$\begin{array}{cccccccccccccc}
 & & & & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \\
 & & & & & & 1 & 1 & 1 & 0 & 1 & 0 & 1 & \\
\hline
 & & & & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \\
 & & & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & & \\
 & & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & & & \\
 & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & & & & \\
 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & & & & & \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & & & & & & \\
\hline
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0
\end{array}$$

$$(x^7+x^6+x^4+x^2+x)\cdot(x^6+x^5+x^4+x^2+1) = x^{13}+x^8+x^7+x^5+x^3+x^2+x.$$

Dividimos por $x^8+x^4+x^3+x+1$.

$$x^{13}+x^{10}+x^8+x^7+x^5+x^3+x^2+x = (x^8+x^4+x^3+x+1)\cdot(x^5+x)+x^7+x^6+x^5+x^4+x^3.$$

Por tanto, $[x^7+x^6+x^4+x^2+x]\cdot[x^6+x^5+x^4+x^2+1] = [x^7+x^6+x^5+x^4+x^3]$.

En notación binaria, $11010110\cdot01110101 = 11111000$, en notación hexadecimal $D6\cdot75 = F8$, mientras que en notación decimal $214\cdot117 = 248$.

Para hallar el resto, podemos, una vez hecha la multiplicación, a la cadena de bits resultante hacer un XOR con la cadena $1000110110\cdots$ las veces necesarias hasta quedarnos con una cadena de 8 bits o menos. Veamos como se haría en este caso:

$$\begin{array}{cccccccccccccc}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
\qquad
\begin{array}{cccccccccc}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\hline
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0
\end{array}$$

Luego el resultado del producto, en notación binaria, es 11111000.

Por último, para calcular el inverso, hemos de aplicar el algoritmo de Euclides. Por ejemplo, vamos a calcular el inverso de $[x^7+x^5+x+1]$. Para ello construimos la siguiente tabla:

				0
				1
$x^8+x^4+x^3+x+1$	x^7+x^5+x+1	$x^6+x^4+x^3+x^2+1$	x	x
x^7+x^5+x+1	$x^6+x^4+x^3+x^2+1$	x^4+x^3+1	x	x^2+1
$x^6+x^4+x^3+x^2+1$	x^4+x^3+1	x^3+x+1	x^2+x	$x^4+x^3+x^2$
x^4+x^3+1	x^3+x+1	x^2	$x+1$	x^5+1
x^3+x+1	x^2	$x+1$	x	$x^6+x^4+x^3+x^2+x$
x^2	$x+1$	1	$x+1$	x^7+x^6+x+1

Por tanto, $[x^7+x^5+x+1]^{-1}=[x^7+x^6+x+1]$, que en notación binaria nos queda $(10100011)^{-1}=11000011$, en notación hexadecimal $A3^{-1}=C3$ y en notación decimal $(163)^{-1}=195$.
Vemos que esta forma de operar en $\mathbb{F}_{256}$ es muy lenta. Para poder trabajar de forma más eficaz vamos a utilzar un resultado de todos los cuerpos finitos.

Definición 11. Sea $K=\mathbb{F}_q$ un cuerpo finito ($q=p^n$). Un elemento primitivo de K es un elemento α que tiene $q-1$ potencias distintas.

Notemos que si α es un elemento primitivo de $\mathbb{F}_q$, entonces los $q-1$ elementos siguientes

$$\alpha^0=1 \quad \alpha^1=\alpha \quad \alpha^2 \quad \cdots \quad \alpha^{q-3} \quad \alpha^{q-2}$$

son todos diferentes, y puesto que α^i nunca puede valer cero (pues de valer cero α sería un divisor de cero), entonces los $q-1$ elementos anteriores son todos los elementos no nulos de $\mathbb{F}_q$. Además, en tal caso, $\alpha^{q-1}=\alpha^0$, y para cualquier $n\in\mathbb{Z}$ se verifica que $\alpha^n=\alpha^{n \bmod q-1}$.
Ejemplo:
Sea $K=\mathbb{F}_{17}=\mathbb{Z}_{17}$. Tomamos el elemento 2, y calculamos sus potencias:

$$2^0=1;\quad 2^1=2;\quad 2^2=4;\quad 2^3=8;\quad 2^4=16;\quad 2^5=15;\quad 2^6=13;\quad 2^7=9;\quad 2^8=1$$

Como tiene 8 potencias distintas, el 2 no es un elemento primitivo. Calculamos entonces las potencias de 3:

$$3^0=1;\quad 3^1=3;\quad 3^2=9;\quad 3^3=10;\quad 3^4=13;\quad 3^5=5;\quad 3^6=15;\quad 3^7=11$$
$$3^8=16;\quad 3^9=14;\quad 3^{10}=8;\quad 3^{11}=7;\quad 3^{12}=4;\quad 3^{13}=12;\quad 3^{14}=2;\quad 3^{15}=6;\ 3^{16}=1$$

Y nos han salido los 16 elementos de $\mathbb{Z}_{17}$ distintos de cero. Por tanto, 3 es un elemento primitivo de $\mathbb{Z}_{17}$.

Teorema 5.6. Todo cuerpo finito tiene al menos un elemento primitivo.

En general, el número de elementos primitivos de $\mathbb{F}_q$ es $\varphi(q-1)$ (salvo para $q=2$). Para el caso $q=17$ se tiene que $\varphi(16)=\varphi(2^4)=2^4-2^3=8$, luego $\mathbb{Z}_{17}$ tiene 8 elementos primitivos. Éstos son: $3,5,6,7,10,11,12,14$.

Notemos que ahora para multiplicar dos elementos en $\mathbb{Z}_{17}$ podemos calcular su logaritmo en base 3 (es decir el exponente al que hay que elevar 3 para que nos de el número, y eso lo podemos hacer viendo las potencias de 3), sumar los logaritmos, reducirlos módulo 16, y elevar 3 al resultado. Por ejemplo:

$$10 \cdot 12 = 3^3 \cdot 3^{13} = 3^{15} = 6 \qquad 15 \cdot 4 = 3^6 \cdot 3^{12} = 3^{18} = 3^2 = 9$$

Mientras que el inverso de 3^x es 3^{16-x}, por ejemplo:

$$14^{-1} = (3^9)^{-1} = 3^{16-9} = 3^7 = 11$$

Esta misma idea es la que vamos a utilizar para la multiplicación y el cálculo de inversos de $\mathbb{F}_{256}$. Vamos a elegir un elemento primitivo, y posteriormente vamos a dar todas sus potencias. Vimos antes que hay un total de 128 elementos primitivos en $\mathbb{F}_{256}$. De todos ellos, el más pequeño es $[x+1]$, 00000011, 03 ó 3. Vamos a escribir una tabla con todas las potencias de $[x+1]$. Para simplificar la notación vamos a usar la representación hexadecimal.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

Supongamos que queremos saber cuanto vale $[x+1]^{125} = (03)^{125}$. Entonces escribimos 125 en hexadecimal, que es 7D. Miramos en la fila precedida por 7 y la columna encabezada por D, y nos que da 20. Entonces $(03)^{125} = 20$ (en notación hexadecimal, luego tendríamos que $[x+1]^{125} = x^5$).

Para construir la tabla, lo mejor es emplear notación binaria. Si tenemos un término de la tabla, vamos a hallar el siguiente. Por ejemplo, a partir de que $(00000011)^{72} = 01100010$. Vamos a calcular $(00000011)^{73}$. Para calcularlo, multiplicamos 01100010 por $[x+1] = 00000011$

$[x+1]^{72}$	$\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$
$x+1$	$\begin{array}{cccccccc} & & & & & & 1 & 1 \end{array}$
$(x+1)^{72}\cdot 1$	$\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$
$(x+1)^{72}\cdot x$	$\begin{array}{cccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$
$(x+1)^{72}\cdot(x+1)$	$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{array}$

Y por tanto, $[x+1]^{73} = 10100110$, que en notación hexadecimal es $A6$.
Si al hacer la multiplicación nos sale una cadena de 9 bits, sumamos $x^8+x^4+x^3+x+1$, y de esta forma la reducimos a una de 8 o menos. Vamos a calcular $[x+1]^{74}$.

$$\begin{array}{cccccccccl} & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & (x+1)^{73} \\ & & & \times & & & & 1 & 1 & x+1 \\ \hline & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & (x+1)^{73}\cdot 1 \\ + & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & (x+1)^{73}\cdot x \\ \hline & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & (x+1)^{73}\cdot(x+1) \\ + & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & x^8+x^4+x^3+x+1 \\ \hline & & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & (x+1)^{74} \end{array}$$

Luego $[x+1]^{74} = [x^7+x^6+x^5+x^4+1]$, que en notación binaria es 11110001, y en hexadecimal F1.
Esta tabla que hemos construido es la tabla de los Antilogaritmos. Nos proporciona lo que vale $[x+1]^y$ para cualquier y comprendido entre 0 y 255.
Vamos a continuación a escribir su tabla inversa, es decir, dado $z \in \mathbb{F}_{256}$, $z \neq 0$ nos da el valor de y tal que $[x+1]^y = z$ (lo que llamaremos $\mathrm{Log}_{x+1}(z)$).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	6E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	74	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	BC	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	B5	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2D	89	B4	7C	B8	26	77	99	E3	A5
F	67	48	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

Que como podemos ver es la inversa de la tabla de los Antilogaritmos.
A partir de estas dos tablas podemos fácilmente multiplicar y calcular inversos.
Vamos a multiplicar, por ejemplo los dos elementos que multiplicamos antes, es decir, vamos a calcular $[x^7+x^6+x^4+x^2+x]\cdot[x^6+x^5+x^4+x^2+1]$.
En representación hexadecimal vamos a multiplicar D6 por 75.
Nos vamos a la tabla de los logaritmos, y nos queda que $\mathrm{Log}_{03}(D6) = 6D$, mientras que $\mathrm{Log}_{03}(75) = 9F$.
Sumamos ambas cantidades y las reducimos módulo 255. Para ello es más fácil expresarlas en decimal. $(109+159)$ mód $255 = 13$.
Y ahora calculamos el Antilogaritmo de $13 = 0D$, que vale F8, lo mismo que nos había dado antes. Es decir, $D6\cdot 75 = F8$.
Aunque si sabemos sumar en hexadecimal, o en binario, podemos hacerlo:

$$\begin{array}{cccc} {\scriptstyle 1} & {\scriptstyle 1} & & \\ & 6 & D & \\ + & 9 & F & \\ \hline 1 & 0 & C & \end{array} \qquad \begin{array}{cccccccc} {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & \\ & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ + & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}$$

Y ahora hemos de reducir módulo FF o módulo 11111111. Lo único que hay que hacer es eliminar el 1 de la izquierda, y sumar 1 al resto (es como si quisiéramos reducir 136 módulo 99. El resultado es 37).

$$10C \to 0C \to 0C+1 = 0D \qquad 100001100 \to 00001100 \to 00001100+1 = 00001101$$

En general, para calcular el producto de dos elementos X e Y podemos proceder como sigue:

$$X\cdot Y = \mathrm{Alog}((\mathrm{Log}_{x+1}(X)+\mathrm{Log}_{x+1}(Y)) \text{ mód } 255)$$

De la misma forma, para calcular el inverso se tiene que:

$$X^{-1} = \mathrm{Alog}(FF-\mathrm{Log}_{x+1}(X))$$

Por ejemplo, $A3^{-1} = \mathrm{Alog}(FF-6F) = \mathrm{Alog}(90) = C3$.

Parte IV

Criptografía Asimétrica

CRIPTOGRAFÍA VIII

Introducción

Uno de los principales problemas que se plantean con los métodos criptográficos es que en la comunicación, tanto emisor como receptor deben compartir la clave que emplean en la transmisión de los mensajes, dado que la clave que se emplea para el cifrado es la misma que se emplea para el descifrado. Esto dificulta enormemente la posibilidad de que dos comunicantes puedan compartir una clave. Transmitir esa clave exigiría que se comunicaran por un canal seguro, pero caso de existir ese canal, ¿que necesidad hay de cifrar la información?. Es necesario para ello establecer una estructura para la distribución de claves, que en general es compleja, y se basa en estructuras jerárquicas.

Suponiendo que hemos conseguido distribuir las claves de forma segura, ¿que ocurriría si alguna clave fuera interceptada por algún atacante?. En ese caso, habría que poner en marcha todo el mecanismo de distribución de claves, pero esto no siempre es posible. Imaginémonos una unidad militar a la que su clave secreta le ha sido interceptada en territorio enemigo.

Otro problema que se plantea es el elevado número de claves. Supongamos que queremos crear una red de comunicación entre 10 nodos. Por cada pareja de nodos es necesaria una clave que sirva para que se comuniquen entre sí. El número total de parejas es 45, lo que exige la distribución de 45 claves. En general, para n nodos serían necesarias $\binom{n}{2} = \frac{n(n-1)}{2}$ claves.

Una solución a estos problemas vino con la *Criptografía de llave pública* o Criptografía asimétrica. En ésta, cada llave consta de dos claves. Una pública, conocida por todo el mundo, y otra privada, que sólo posee una persona. De esta forma no es necesario que emisor y receptor compartan una clave secreta.

El inicio de la criptografía de llave pública podemos encontrarlo en noviembre de 1976, cuando los investigadore de Stanford Whitfield Diffie y Martin Hellman publicaron el trabajo *New Directions in Cryptographi* en la revista IEEE Transactions on Information Theory. Los autores hacen referencia al trabajo de Merkle, de la Universidad de California, quien había llegado a la misma idea.

En su trabajo, Diffie y Hellman presentan las bases de la Criptografía de llave pública, plantean algunos problemas que se pueden resolver con ella, y describen un protocolo que permite el intercambio de cierta información a través de un canal inseguro.

Sin embargo, aunque se considera a estos tres investigadores como los padres de la criptografía asimétrica, parece ser que ésta ya fue desarrollada previamente por organizaciones gubernamentales dedicadas a la criptografía.

Así, a finales de los 70, Robert Inman, entonces director de la NSA, manifestó que una década antes de la publicación del artículo de Diffie y Hellman, la agencia ya hacía uso de la criptografía asimétrica. Aunque no mostró ningún documento en que basar su afirmación, varias personas vinculadas a la alta seguridad estadounidense han confirmado este hecho. Incluso se vincula el interés de la NSA por la criptografía de llave pública con el control del armamento nuclear. Por otra parte, en 1997, el GCHQ (Goverment Communication Headquartes), equivalente británico de la NSA, desclasificó una serie de documentos que muestran que dicha organización trabajaba con la idea de la llave pública a finales de los 60.

1 Requisitos de la criptografía asimétrica

Como hemos dicho, Diffie y Hellman publicaron en su trabajo lo que pensaban que serían las bases de la criptografía de llave pública.
Supongamos que A y B son dos comunicantes que van a usar un sistema de cifrado asimétrico. Para ello, hay un par de algoritmos públicos, E y D (que podrían coincidir), uno para cifrar y otro para descifrar. Ambos algoritmos se usan con una clave de cifrado (pública) y una clave de descifrado (secreta).
Los requisitos antes referidos son entonces:

1. Cada comunicante, A y B, pueden calcular en poco tiempo sus parejas de claves: (e_a, d_a) para A y (e_b, d_b) para B. Las claves e_a y e_b son públicas, mientras que d_a y d_b se mantienen secretas.

2. El comunicante A, conociendo la clave pública de B, puede calcular fácilmente el cifrado de un mensaje M usando el algoritmo de cifrado E. Este mensaje cifrado será $C = E_{e_b}(M)$.

3. El comunicante B puede, a partir de su clave secreta, recuperar $M = D_{d_b}$ en poco tiempo.

4. El cálculo de d_a o d_b a partir del conocimiento de e_a o e_b es computancionalmente intratable.

5. El cálculo de M a partir de C y e_b requiere también la realización de calculos de una complejidad muy alta, de forma que dicho cálculo es prácticamente irrealizable.

La existencia de sistemas de cifrado asimétrico están ligadas de las denominadas **funciones unidireccionales con trampa**
Una función unidereccional es una función f, de forma que el cálculo de la imagen de un elemento x, es decir, el cálculo de $f(x)$ es viable, mientras que a partir del conocimiento de $f(x)$ es computacionalmente muy costoso el cálculo de x.
Una función unidireccional con trampa es una función unidireccional en la que existe una información adicionasl (la trampa) que permite el cálculo de x a partir del conocimiento de $f(x)$ (es decir, se puede calcular $f^{-1}(y)$).

Más precisamente, si f es una función biyectiva (en principio bastaría que fuera inyectiva), para ser una función unidireccional con trampa se le exigen las siguientes condiciones:

1. El cálculo de la función directa $y = f(x)$ puede realizarse en un tiempo razonable. Esta condición se traduce en el cumplimiento del requisito 2 para un sistema de cifrado asimétrico.

2. Existe una información adicional cuyo conocimiento permite el cálculo de la función inversa $x = f^{-1}(y)$ en un tiempo razonable. Esta información adicional sería la clave secreta. Con esto, se cumple el requisito tercero dado para un cifrado asimétrico.

3. Sin el conocimiento de la información adicional, resulta prácticamente imposible el cálculo de $x = f^{-1}(y)$ aún disponiendo de una gran potencia de cálculo. Es decir, sin la información adicional, la función f es unidireccional. Esto se traduce en la condición del requisito quinto.

4. El cálculo de la información adicional resulta totalmente inviable. Con esto nos aseguramos que se satisface el cuarto requisito.

En su artículo, Diffie y Hellman publicaron también un protocolo para intercambio de información por un canal inseguro. Veámoslo a continuación:

2

Protocolo de Intercambio de Diffie-Hellman

La seguridad de este protocolo se basa en la dificultad del cálculo de logaritmos. Para llevarlo a la práctica, necestamos un grupo $(G, \cdot)$ y α un generador de G (es decir, un elemento de G cuyas potencias son todos los elementos de G). Si el elemento α no fuera un generador, pero el número de potencias suyas fuera elevado, también podría valer. Normalmente, el grupo G se elige como el conjunto de los elementos distintos de 0 de $\mathbb{Z}_p$, con p un número primo lo suficientemente grande.
Supongamos que dos personas A y B desean intercambiar una información secreta. Entonces pueden proceder como sigue:

1. A elige un número al azar x_a, comprendido entre 2 y el número de elementos de G menos 2.

2. A calcula $\alpha^{x_a} = y_a$ y envía esta información a B.

3. A su vez, B elige otro número al azar x_b y envía a A el resultado de calcular $y_b = \alpha^{x_b}$.

4. A calcula $z_{ba} = y_b^{x_a}$.

5. B calcula $z_{ab} = y_a^{x_b}$.

Puesto que $z_{ab} = z_{ba}$, ya que

$$z_{ab} = y_a^{x_b} = (\alpha^{x_a})^{x_b} = \alpha^{x_a \cdot x_b} = \alpha^{x_b \cdot x_a} = (\alpha^{x_b})^{x_a} = y_b^{x_a} = z_{ba},$$

entonces A y B comparten dicha información secreta.
Por el canal únicamente han viajado los datos y_a e y_b. El conocimiento de ambos valores no permite calcular x_a ni x_b, por la dificultad del cálculo del logaritmo, luego no es posible calcular $z_{ab} = z_{ba}$.
El inconveniente de este protocolo es que ni A ni B pueden elegir la información que van a compartir, pues dicha información depende de dos valores x_a y x_b sobre los que uno de los comunicantes no puede ejercer control alguno.
Vamos a ver algún ejemplo de intercambio de información.
Tomamos $p = 27883$. Un generador de $\mathbb{Z}_{27833}^*$ es $\alpha = 2339$. Ambos elementos son conocidos por A y B.
A elige un número x_a entre 2 y 27881, por ejemplo $x_a = 15343$ y calcula $y_a = 2339^{15343} (\text{mód } 27883)$.
El resultado es $y_a = 10679$. Este resultado se lo envía a B.
Por otra parte, B elige otro número, por ejemplo, $x_b = 21955$ y realiza el mismo cálculo: $y_b = 2339^{21955} (\text{mód } 27883) = 11673$. Envía a A dicho valor.
Ahora A calcula $y_b^{x_a} = 11673^{15343} (\text{mód } 27883) = 19612$, mientras que B calcula $y_a^{x_b} = 10679^{21955} (\text{mód } 27883) = 19612$. Y por tanto, A y B comparten esa información.
Un posible ataque a este algoritmo viene cuando el atacante, C, no sólo escucha la conversación, sino que toma partido en ella. El atacante podría interceptar las comunicaciones entre A y B, y modificarlas a su antojo.
En tal caso, C genera un número aleatorio. x_c
Cuando A envíe el valor $y_a = \alpha^{x_a}$, C lo intercepta y envía a B el valor $y_c = \alpha^{x_c}$, haciéndose pasar por A.
De la misma forma, cuando B envíe a A el valor $y_b = \alpha^{x_b}$, C lo intercerpta y envía a A $y_c = \alpha^{x_c}$.
A, que ha recibido y_c creyendo que ha venido de B calcula $z_{ca} = y_c^{x_a}$. Por su parte, C que disponde del dato y_a calcula $z_{ac} = y_a^{x_c}$. A y C comparten el valor $z_{ca} = z_{ac}$, aunque A cree que lo comparte con B.
Por su parte, B que posee también el dato y_c puede calcular $z_{cb} = y_c^{x_b}$, mientras que C puede calcular $z_{bc} = y_b^{x_c}$. Ahora B y C comparten $z_{cb} = z_{bc}$, aunque B cree que lo comparte con A.
Si por ejemplo, la información que A y B es una clave que van a emplear para estableces una comunicación usando un sistema de cifrado simétrico, el atacante podría interceptar toda la información que se envíen, descifrarla y modificarla a su antojo, sin que A ni B se percaten de la acción de este intruso.
Más adelante veremos algunas soluciones a este tipo de ataques (ataques activos, pues el atacante interviene de forma activa en la comunicación). Sin embargo, por ahora nos vamos a ocupar únicamente de ataques pasivos, en los que el atacante únicamente se limita a escuchar la comunicación, pero sin intervenir en ella.

CRIPTOGRAFÍA IX

El criptosistema RSA

Este es, sin lugar a dudas, el sistema de cifrado asimétrico más conocido y también el más empleado. En agosto de 1977 apareció en la revista *Scientific American* una breve descripción de este criptosistema. El nombre proviene de las iniciales de los apellidos de sus creadores: Ronald River, Adi Shamir y Leonard Aldeman, investigadores del *Massachussettts Institute of Technology*. A la NSA, interesada por el algoritmo, no le gustó esta publicación, y trató de prohibir la publicación de los detalles del mismo. Sus intentos fueron en vano, y en febrero de 1978 se produjo esta publicación. Hubo entonces una intensa batalla legal con el gobierno de los Estados Unidos, hasta que en 1982 los tres creadores fundaron la compañía RSA Data Security, con el fin de comercializar las aplicaciones de su criptosistema. En 1996 la vendieron por 200 millones de dólares.

1 Descripción del algoritmo

La fortaleza del RSA se basa en el problema de la factorización de números enteros como producto de primos.
Lo primero que hay que elegir es elegir las claves. Como en todo sistema de cifrado asimétrico, las claves van por parejas: pública y privada.

1.1 *Elección de las claves*

Para elegir la pareja de claves, lo primero que necesitamos es elegir dos primos grandes p y q, y calcular su producto $n = p \cdot q$.
Ahora se elige un entero e tal que $mcd(e, \varphi(n)) = 1$. Recordemos que $\varphi(n) = (p-1)\cdot(q-1)$. Ambos valores, (n, e) constituyen la clave pública del criptosistema. El número n se conoce como *módulo del Criptosistema*.
La clave privada es un entero d tal que $d \cdot e \equiv 1 (\text{mód } \varphi(n))$.
Ejemplo:
Vamos a dar dos ejemplo: uno con números pequeños y otros con primos grandes.

1. Sea $p = 383$ y $q = 881$. Estos valores se mantienen en secreto. Calculamos n que vale $383 \cdot 881 = 337423$. En tal caso se tiene que $\varphi(n) = 336160$.

Elegimos un número e que sea primo relativo con 336160, por ejemplo $e = 17$.

La clave pública es entonces $(337423, 17)$.

Para hallar la clave privada resolvemos la congruencia

$$17d \equiv 1 (\text{mód } 336160)$$

es decir, calculamos 17^{-1} en $\mathbb{Z}_{336160}$, que podemos hacerlo usando el algoritmo INVERSO que viene en el apéndice del final del capítulo.

Dicho inverso vale 158193. La clave privada es entonces 158193.

2. Tomamos ahora

$p = 10812981046982860638137379673045680314065226768577395553398805175629252215305585242965995842861637519087133648293907956480741461975507825249009631752637576032197$

$q = 20461645447532839139961913561561538563680845596311680282072992740226063562164517724836427209397774783960112596186378507367196150974918934877794517781 1$

Su producto $n = p \cdot q$ vale:

$22125138441425572298487804846766830279864046004221720939401792778570835195751003077640454321551835347750302308372508242946726255443077886620078472 08$

$21024893019995202322829392620136980238846778610444136837941077029200210413168118438406093673856005835976216379773432614899494391940398116802412376672208140973609$

Con estos números, $\varphi(n) = (p-1) \cdot (q-1)$ vale:

$22125138441425572298487804846766830279864046004221720939401792778570835195751003077640454321551835347750302308372508242946726255443077886620078472 08$

$19943594915092299804540326261432793071824640547949588826638079708816555264235299278487848912321477811973525267104440693289556460669175372767762224148166438192580$

Nótese que las cifras de n y $\varphi(n)$ coinciden en la fila superior de ambos números.

Se puede comprobar que $n = 2^{1024} \cdot 1'001110110001...)_2$. Es decir, la expresión de n en binario tiene 1025 cifras, y las más significativas son $1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1$

Elegimos ahora el número $e = 65537$, y calculamos d de forma que $65537d \equiv 1 (\text{mód } \varphi(n))$. El valor de d resulta ser:

$d = 26244842797754306582303766556108051725843888740226079708090015878146645835446892278495642445602326623649366029157251488574065018205668170740877678557978569$
$46259254373377628662215814781482772717121259904387268919322428400177251348995169961158410010391507370533393231227601220570429176416217683950330112593663 3$

. . . . . 1.2

Cifrado y descifrado

Supongamos que tenemos un mensaje m que queremos enviar al poseedor de un criptosistema RSA de clave pública (n, e) y clave privada d, y que dicho mensaje deseamos enviarlo cifrado con este criptosistema. Para esto, el mensaje debe ser primo relativo con n y menor que n. De ser mayor que n, habría que dividir el mensaje m en varios submensajes $m_0, m_1, \cdots, m_k$ expresando el número m en base n. Es decir, hacemos

$$m = m_0 + m_1 \cdot n + \cdots + m_k n^k$$

con $m_k < n$. Entonces lo que se cifraría sería cada uno de los submensajes m_i.
El cifrado de un mensaje m (o de los mensajes m_i) se haría con la clave pública (n, e) como sigue:

$$E_K(m) = c = m^e (\text{mód } n)$$

De ahí que al número e se le llame *Exponente de cifrado*.
Recibido un criptograma, se descifraría con la clave privada realizando una operación análoga a la de cifrado:

$$D_K(c) = m' = c^d (\text{mód } n)$$

Y por tanto, al número d se le llama *exponente de descifrado*.
Vamos a ver a continuación que el proceso de descifrado es inverso al de cifrado. Dicho de otra forma, para un mensaje m se verifica que $D_K(E_K(m)) = m$.

Partimos de un mensaje m que verifica que $m < n$ y $mcd(m, n) = 1$.

Puesto que $mcd(m, n) = 1$, por el teorema de Fermat se tiene que $m^{\varphi(n)} \equiv 1(\text{mód } n)$, y dado que $e \cdot d \equiv 1(\text{mód } \varphi(n))$ existe un número entero k tal que $e \cdot d = k \cdot \varphi(n) + 1$. En estas condiciones podemos ver que:

$$D_K(E_K(m)) = D_K(m^e) = (m^e)^d = m^{e \cdot d} = m^{\varphi(n) \cdot k + 1} = m^{\varphi(n) \cdot k} \cdot m = (m^{\varphi(n)})^k \cdot m = 1^k \cdot m = m$$

Donde todas las igualdades están tomadas módulo n.

¿Que ocurriría si $mcd(m, n) \neq 1$?. En este caso, el proceso de descifrado es inverso al de cifrado.

Pueden darse tres situaciones:

- $m = 0$. En este caso, el mensaje cifrado es también 0. Este mensaje carece absolutamente de interés por razones obvias.
- $mcd(m, n) = p$. Vamos a ver aquí que $m^{e \cdot d} \equiv m(\text{mód } p)$ y que $m^{e \cdot d} \equiv m(\text{mód } q)$. De ser así, por el teorema chino del resto tendríamos que $m^{e \cdot d} \equiv m(\text{mód } n)$.

  - Puesto que $mcd(m, n) = p$ se tiene que $p|m$, y por tanto, $m \equiv 0(\text{mód } p)$, luego $m^{e \cdot d} \equiv 0(\text{mód } p)$.

  - Se tiene que $mcd(m, q) = 1$. Notemos que $e \cdot d = k \cdot \varphi(n) + 1 = d \cdot (p-1) \cdot (q-1) + 1 = k' \cdot (q-1) + 1$.

  Procediendo de igual forma que antes, se tiene que $m^{e \cdot d} = m$ en $\mathbb{Z}_q$, es decir, $m^{e \cdot d} \equiv m(\text{mód } q)$.
- $mcd(m, n) = q$. Aquí se razona igual que en el caso precedente.

Si para cualquier mensaje m se verifica que $D_K(E_K(m)) = m$, ¿a qué viene la restricción de que $mcd(m, n) = 1$?. El problema viene con un mensaje m que no sea primo relativo con n (y que no sea nulo) se podría romper el criptosistema (tanto con el mensaje como con el criptograma). El motivo es que, si $mcd(m, n) = p$, conociendo m, o conociendo c se podría calcular $p = mcd(m, n) = mcd(c, n)$ (el valor n es público), y a partir de ahí se calcula q, y una vez hecho esto es fácil obtener la clave privada.

¿Cómo puede controlar el poseedor de un criptosistema RSA que nadie va a querer enviarle un mensaje m que no sea primo relativo con n?. La respuesta es que en principio no puede hacer nada. Sin embargo, la probabilidad de que eso ocurra es muy pequeña.
El número posible de mensajes es n. De ellos, hay $\varphi(n)$ que son primos relativos con n, luego hay $n - \varphi(n)$ que no lo son. La probabilidad de dar con uno de ellos es

$$\frac{n-\varphi(n)}{n} = \frac{p\cdot q-(p-1)\cdot(q-1)}{p\cdot q} = \frac{p\cdot q-p\cdot q+p+q-1}{p\cdot q} = \frac{1}{p}+\frac{1}{q}-\frac{1}{p\cdot q} \approx \frac{1}{p}+\frac{1}{q}$$

Ejemplo:
Vamos a continuar con los dos ejemplos que vimos anteriormente.

1. Vamos a cifrar, con la clave pública $(337423, 17)$ el mensaje $m = 123456$. Para eso, calculamos $123456^{17}(\text{mód } 337423)$, y nos sale 184183. Ese sería por tanto el mensaje cifrado.

   Para descifrarlo, necesitamos la clave privada, es decir, 158193. Por tanto, el criptograma hemos de elevarlo a ese número, y se tiene que $184183^{158193} \equiv 123456(\text{mód } 337423)$, como era de esperar.

   Vamos a cifrar ahora el mensaje $m = 97791$. Notemos que $mcd(97791, 337423) = 881$, y por tanto $97791^{336160} \not\equiv 1(\text{mód } 337423)$. De hecho, $97791^{336160} \equiv 8810(\text{mód } 337423)$.

   Calculamos $E_K(m) = 97791^{17}(\text{mód } 337423) = 273991$.

   Para descifrarlo hallamos $273991^{158193}(\text{mód } 337423)$, y el resultado es 97791.

   Si alguien interceptara el mensaje cifrado $c = 273991$ podría calcular $mcd(273991, 337423)$ y obtendría 811, y a partir de ahí podría determianr fácilmente el valor de d.

   Vamos a determinar la probabilidad de dar con un mensaje que no sea primo relativo con n. Para esto, el mensaje deber ser múltiplo de 383 (hay 881 posibilidades, desde $0 \cdot 383$ hasta $880 \cdot 383$) o múltiplo de 881, para lo que hay 383 posibilidades. Luego en total hay $881 + 383 - 1 = 1263$ (pues el cero lo hemos contado dos veces), y la probabilidad es entonces $\frac{1263}{337423} = 0'003743$, mientras que $\frac{1}{p} + \frac{1}{q}$ vale $0'003476$.

2. Vamos a trabajar con el otro criptosistema RSA que describimos en el ejemplo anterior.

   Tomamos el mensaje para cifrar

   $m = 12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890$

   Para cifrarlo, hallamos $c = m^{65537}(\text{mód } n)$, y el resultado es

   $c = 630482394740207612685151828890063508526442610996049631624341483464978881591080185236759269497428538765108573056805441614793969441494342110195943536025090712205065932099380853722913860415672694228330994624537146063433754149678366054051762062311430692985539246114754288082867205741242030296350551006469850258995$

   El cálculo de c^d nos da como resultado m.

   De entre los n posibles mensajes, hay $n - \varphi(n)$ mensajes que no son primos relativos con n. El valor de $n - \varphi(n)$ es

   $10812981049029025182890663587041871670221380624945480113029973203836551489328191599182447615345280240026911126689919216099379312712227440346501525240417027810299$

es decir, $n - \varphi(n) \approx 1'0821 \cdot 10^{159}$, mientras que $n \approx 2'2125 \cdot 10^{308}$. La probabilidad de dar con un mensaje que no sea primo relativo con n es el cociente entre ambos valores, que es aproximadamente igual a $4'88719249267 \cdot 10^{-150}$ (muchísimo más fácil es sentarnos en un millón de pajares, y clavarnos en cada uno de ellos la aguja).

. . . . . 1.3

Precauciones en la elección de los parámetros

Hemos visto hasta ahora el funcionamiento general de un criptosistema RSA. Pero si no se toman determinadas precauciones en la elección de los primos, o del exponente de cifrado, el sistema podría tener importantes debilidades. Veamos aquí algunas de ellas, y como solucionarlas.

Números primos próximos entre sí

Es claro que si se consigue factorizar el parámetro n de un criptosistema RSA este queda roto. Por tanto, habrá que elegir los primos p y q de forma que no sea posible encontrar dicha factorización en un tiempo razonable.
En el apéndice del final del capítulo se describe un método de factorización que es efectivo cuando los números primos están próximos entre sí. Este método consiste en encontrar un número x tal que $x^2 - n$ sea cuadrado perfecto. Se comienza con x el entero inmediatamente superior a $\sqrt{n}$, y se va sumando 1 hasta lograr que $x^2 - n$ sea cuadrado perfecto.
Por ejemplo, para el caso $n = 337423$ comenzaríamos probando con $x = 581$, $x = 582$, y así hasta llegar a $x = 632$, pues $632^2 - 337423 = 249^2$. En tal caso, la factorización de n sería $(632 + 249) \cdot (632 - 249)$.
En el caso que nos ocupa sería necesario hacer 52 comprobaciones para llegar a la factorización.
Si tomamos el otro criptosistema RSA descrito, el número de comprobaciones necesarios para llegar a la factorización de n por este método sería
540634177930134112305908837741393867917453860124108494132892860062283519588961662413501018391182877343552038354828214367854548025344065637245441258972089395377
es decir, $5'4 \cdot 10^{158}$ comprobaciones. Si tuviéramos máquinas capaces de hacer un billón de comprobaciones por segundo, y estuvieran trabajando 1000000 de máquinas a la vez desde el inicio del universo, llevaríamos ahora menos de 10^{35} comprobaciones hechas.

Claves parejas

Hemos visto que lo que hace que los procesos de cifrado y descifrado sean inverso uno del otro es consecuencia de que para m tal que $mcd(m, n) = 1$ se verifica, en virtud del teorema de Fermat, que $m^{\varphi(n)} \equiv 1 (\text{mód } n)$.
Pero esto también se podría haber deducido a partir del teorema pequeño de Fermat y el teorema chino del resto.
Si $mcd(m, n) = 1$, entonces $mcd(m, p) = 1$ y $mcd(m, q) = 1$. Por tanto, se tiene:

$$m^{p-1} \equiv 1 (\text{mód } p) \Longrightarrow m^{(p-1)\cdot(q-1)} \equiv 1 (\text{mód } p) \quad \text{es decir,} \quad m^{\varphi(n)} \equiv 1 (\text{mód } p)$$
$$m^{q-1} \equiv 1 (\text{mód } q) \Longrightarrow m^{(q-1)\cdot(p-1)} \equiv 1 (\text{mód } q) \quad \text{es decir,} \quad m^{\varphi(n)} \equiv 1 (\text{mód } q)$$

de donde deducimos que $m^{\varphi(n)} \equiv 1(\text{mód } n)$. Pero si analizamos esto con más detalle vemos que lo que necesitamos realmente es que el exponente de cifrado sea múltiplo de $p-1$ y $q-1$ (algo que obviamente, $\varphi(n) = (p-1)\cdot(q-1)$ lo cumple), y esto se puede conseguir con otros exponentes, en concreto con $\alpha = mcm(p-1, q-1)$.
Dado que $mcm(p-1, q-1) = \frac{(p-1)\cdot(q-1)}{mcd(p-1,q-1)}$ y $mcd(p-1, q-1) \geq 2$ (pues ambos son pares), entonces $mcm(p-1, q-1) \leq \frac{\varphi(n)}{2}$. Por tanto, para un exponente de cifrado hay al menos dos posibles exponentes de descifrado.
Ejemplo:
Para $p = 383$ y $q = 881$ tenemos que $mcm(p-1, q-1) = \frac{(p-1)\cdot(q-1)}{2} = 168080$ (es decir, el mejor caso posible). Entonces para $e = 17$ tenemos dos posibles exponentes de descifrado. Basta resolver la congruencia $17d \equiv 1(\text{mód } 168080)$, cuya solución es $d = 158193 + 168080 \cdot k$. Por tanto, se puede tomar $d = 158193$ y $d = 326723$.
Pero si tomamos $n = 293383$ entonces $p = 397$, $q = 739$, $\varphi(n) = 292248$, entonces $mcd(p-1, q-1) = 18$, luego $mcm(p-1, q-1) = \frac{292248}{18} = 16236$. Tomamos por ejemplo $e = 1249$. Entonces, el valor que nos sale para d es $d = 81193$ (d viene de resolver la congruencia $1249d \equiv 1(\text{mód } 292248)$). Pero si resolvemos la congruencia $1249d \equiv 1(\text{mód } 16236)$ nos sale la solución $d = 13 + 16236 \cdot k$. Dando valores a k nos salen los siguientes exponentes de descifrado:

$$\left\{ \begin{array}{cccccccccc} 13 & 16249 & 32485 & 48721 & 64597 & 81193 & 97429 & 113665 & 129901 & 146137 \\ 162373 & 178609 & 194845 & 211081 & 227317 & 243553 & 259789 & 276025 & 292261 & 308497 \end{array} \right\}$$

En la última ya hemos superado el valor de n, luego ese exponente no se tendría en cuenta. De hecho, tampoco es significativo el penúltimo, pues es mayor que $\varphi(n)$. Nos quedan entonces 18 exponentes de descifrado, exactamente el valor de $mcd(p-1, q-1)$.
Por tanto, lo que hay que procurar es que $mcd(p-1, q-1)$ sea lo más pequeño posible (sabemos que como mínimo vale 2). Esto puede conseguirse buscando primos p y q de forma que $\frac{p-1}{2}$ y $\frac{p-1}{2}$ sean también primos (aunque esta condición no es necesaria, pues el número $\frac{881-1}{2} = 440$ no es primo, y sin embargo con los primos 383 y 880 teníamos únicamente dos exponentes de descifrado.

Mensajes no cifrables

Dado un criptosistema RSA de llave pública (n, e) y llave privada d, un mensaje m se dice no cifrable si $E_K(m) = m$. La existencia de tales mensajes es inevitable (por ejemplo, $m = 1$). Sin embargo, vamos a tratar de minimizar la cantidad de tales mensajes.
Para que un mensaje m no sea cifrable es necesario que se cumpla que $m^e \equiv m(\text{mód } n)$, o lo que es lo mismo.

$$m^e \equiv m(\text{mód } p) \qquad m^e \equiv m(\text{mód } q)$$

Lo que hemos de ver es cuántas soluciones tiene la ecuación $x^e = x$ en $\mathbb{Z}_p$. El número mínimo de soluciones es 3: 0, 1 y $p-1$ (esta última puesto que e tiene que ser impar para ser primo

relativo con $\varphi(n)$. Puesto que $x^e - x = x \cdot (x^{e-1} - 1)$, hemos de ver cuantas soluciones tienen las ecuaciones $x = 0$ y $x^{e-1} = 1$.
La primera, obviamente tiene una única solución.
El número de soluciones de la segunda es $mcd(e-1, p-1)$. Si y es un elemento primitivo de $\mathbb{Z}_p$, estas soluciones son:

$$\{1 = y^{\frac{0\cdot(p-1)}{d}},\ y^{\frac{(p-1)}{d}}, y^{\frac{2\cdot(p-1)}{d}},\ \cdots,\ y^{\frac{(d-1)\cdot(p-1)}{d}}\}$$

(la solución -1 se obtiene como $y^{\frac{\frac{d}{2}\cdot(p-1)}{d}} = y^{\frac{p-1}{2}}$).
Por tanto, el número de soluciones de $m^e \equiv m(\text{mód } p)$ es $1 + mcd(e-1, p-1)$, luego el número de mensajes no cifrables es

$$(1 + mcd(e-1, p-1)) \cdot (1 + mcd(e-1, q-1))$$

luego hay siempre un mínimo de 9 mensajes no cifrables.
Por ejemplo, vamos a tomar $p = 383$, $q = 881$, y nuestro exponente de cifrado va a ser $e = 17$. En este caso, $mcd(e-1, p-1) = 2$, mientras que $mcd(e-1, p-1) = 16$. Por tanto, el número de mensajes no cifrables es $3 \cdot 17 = 51$. Vamos a calcularlos:
En $\mathbb{Z}_{383}$ la ecuación $x^{16} = 1$ tiene dos soluciones, que son 1 y 382.
En $\mathbb{Z}_{881}$ tenemos 16 soluciones. Para hallarlas, tomamos un elemento primitivo de $\mathbb{Z}_{881}$, por ejemplo $y = 3$, y las soluciones son:

$$\{y^0,\ y^{55},\ y^{110},\ \cdots,\ y^{825}\}$$

es decir,

$$\{1,\ 767,\ 662,\ 298,\ 387,\ 813,\ 704,\ 796,\ 880,\ 114,\ 219,\ 583,\ 494,\ 68,\ 177,\ 85\}$$

Las 51 soluciones se obtienen resolviendo los 51 sistemas de congruencias:

$$\begin{aligned} m &\equiv a(\text{mód } 383) \\ m &\equiv b(\text{mód } 881) \end{aligned}$$

donde a es solución de $x^{17} \equiv x(\text{mód } 383)$ y b es solución de $x^{17} \equiv x(\text{mód } 881)$.
Los 51 elementos son:

$$\left\{\begin{array}{rrrrrrrrr} 0 & 1 & 767 & 8043 & 8809 & 8810 & 16853 & 17619 & 26043 \\ 34853 & 43663 & 65109 & 65492 & 67024 & 73919 & 74302 & 75834 & 82729 \\ 83112 & 84644 & 86557 & 95367 & 104177 & 119112 & 127922 & 136732 & 161501 \\ 170311 & 179121 & 200691 & 209501 & 218311 & 252779 & 254311 & 254694 & 261589 \\ 263121 & 263504 & 270399 & 271931 & 272319 & 293760 & 302570 & 311380 & 319804 \\ 320570 & 328614 & 329380 & 328613 & 336656 & 337422 & & & \end{array}\right\}$$

Una posible solución para evitar esto es tomar como exponente de cifrado $e = 3$ (en el caso de que $mcd(3, \varphi(n)) = 1$). Pero un exponente de cifrado tan pequeño puede dar lugar a otros problemas de seguridad que veremos más adelante.

Otra solución es una que nos ha aparecido en el apartado anterior. Elegir primos p y q tales que $\frac{p-1}{2}$ y $\frac{q-1}{2}$ sean también primos. En tal caso, eligiendo e al azar, las únicas posibilidades son $mcd(e-1, p-1) = 2$ y $mcd(e-1, p-1) = \frac{p-1}{2}$, y lo mismo con respecto a q. La probabilidad de que se de la segunda opción es muy pequeña.
También se suele elegir como exponente de cifrado $65537 = 2^{16} + 1$, con el cual se facilita el cifrado.

Exponente de cifrado común

Supongamos que unos cuantos abonados tienen el mismo exponente de cifrado e. Entonces un mensaje, enviado un mínimo de e veces puede ser descubierto sin necesidad de conocer la clave de descifrado.
Concretemos un poco. Supongamos que tenemos r criptosistemas RSA de claves públicas $(n_1, e), (n_2, e), \cdots, (n_r, e)$, en la que no se ha empleado ningún número primo dos veces (lo que implica que $mcd(n_i, n_j) = 1$) con las correspondientes claves privadas $d_1, d_2, \cdots d_r$. Supongamos también $e \leq r$ y que se envía un mensaje m cifrado con las anteriores claves públicas. Sean $c_1, c_2, \cdots, c_r$ los correspondientes criptogramas, es decir,

$$c_1 = m^e (\text{mód } n_1) \qquad c_2 = m^e (\text{mód } n_2) \qquad \cdots \qquad c_r = m^e (\text{mód } n_r)$$

En tal caso, el sistema de congruencias

$$\begin{aligned} x &\equiv c_1 (\text{mód } n_1) \\ x &\equiv c_2 (\text{mód } n_2) \\ &\cdots\cdots\cdots\cdots \\ x &\equiv c_e (\text{mód } n_e) \end{aligned}$$

tiene como solución a m^e. Como además $m^e < n_1 n_2 \cdots n_e$, la menor solución positiva del anterior sistema es m^e.
Construimos ahora un criptosistema RSA de llave pública (n', e), con la condición de que $(n')^e > m^e$, y "desciframos" el mensaje m^e con ese criptosistema. El resultado final es m.
Ejemplo:
Vamos a considerar tres criptosistemas RSA con llaves públicas $(337423, 3)$, $(290891, 3)$ y $(337237, 3)$. Con esos tres criptosistemas vamos a cifrar el mensaje $m = 123321$. Los correspondientes criptogramas que obtenemos son:

$$c_1 = 270656 \qquad c_2 = 79162 \qquad c_3 = 140499$$

Supongamos que alguien intercepta estos tres valores. Con el conocimiento que tiene de las llaves públicas podría plantear el sistema de congruencias:

$$\begin{aligned} x &\equiv 270656 (\text{mód } 337423) \\ x &\equiv \ \ 79162 (\text{mód } 290891) \\ x &\equiv 140499 (\text{mód } 337237) \end{aligned}$$

que es equivalente a $x \equiv 1875474282205161 (\text{mód } 33100929117333641)$. Por tanto, tendría que $m^3 = 1875474282205161$. A partir de aquí necesita hallar una raíz cúbica de este número. Esto puede hacerse fácilmente tomando una estimación de dicha raíz cuadrada, y luego aproximando por el método de Newton-Raphson o cualquier otro método para determinar raíces de una función.
Una primera estimación puede obtenerse teniendo en cuenta que $m^3 \approx 1'875 \cdot 10^{15}$. Puesto que la raíz cúbica de $1'876$ vale $1'23332$, una estimación al alza de la raíz cúbica sería $1'23332 \cdot 10^5 = 123332$.
En este caso, el proceso de Newton-Raphson partiría de $x_0 = 123332$ y construría la sucesión con la regla

$$x_{n+1} = \frac{2x_n^3 + 1875474282205161}{3x_n^2}.$$

La sucesión resultante sería:

$$x_0 = 123332 \qquad x_1 = 123321,000981062519572 \qquad x_2 = 123321,000000000007805$$

que permite ver que $\sqrt[3]{1875474282205161} = 123321$.
También podía hallar la raíz cúbica como hemos explicado más arriba. Elige un criptosistema RSA de clave pública $(n', 3)$, con $(n')^3 > m^3$ y de forma que $mcd(\varphi(n'), 3) = 1$. Por ejemplo, puede elegir $n' = 515197 = 677 \cdot 761$. En tal caso, $\varphi(n') = 513760$, y d sería la solución de la congruencia $3d \equiv 1 (\text{mód } 513760)$, es decir, $d = 342507$.
Puesto que $1875474282205161 (\text{mód } 515197) = 144551$, la raíz cúbica de 1875474282205161 vendrá dada por $144551^{342507} (\text{mód } 515197)$. El resultado, como era de esperar, es 123321.
Para evitar problemas de este tipo es para lo que se aconseja no usar exponentes de cifrado muy bajos.

Primos comunes

Lo que tenemos ahora son r usuarios con criptosistemas RSA, todos ellos con el mismo módulo de cifrado. Es decir, sus claves públicas son (n, e_1), (n, e_2), $\cdots$, (n, e_r), y suponemos que $mcd(e_1, e_2, \cdots, e_r) = 1$.
En tal caso, un mismo mensaje m enviado a estos usuarios podría ser descifrado por un atacante que interceptara los distintos criptogramas.
Para eso, el atacante deberá encontrar números enteros $s_1, s_2, \cdots, s_r$ tales que $s_1 \cdot e_1 + s_2 \cdot e_2 + \cdots + s_r \cdot e_r = 1$. Estos elementos pueden encontrarse usando el algoritmo extendido de Euclides.
A partir de esto, si $c_1, c_2, \cdots, c_r$ son los criptogramas enviados, es posible recuperar el mensaje m sin más que realizar el siguiente cálculo en $\mathbb{Z}_n$:

$$c_1^{s_1} \cdot c_2^{s_2} \cdots c_r^{s_r}$$

pues

$$c_1^{s_1} \cdot c_2^{s_2} \cdots c_r^{s_r} = m^{e_1 \cdot s_1} \cdot m^{e_2 \cdot s_2} \cdots m^{e_r \cdot s_r} = m^{s_1 \cdot e_1 + s_2 \cdot e_2 + \cdots + s_r \cdot e_r} = m^1 = m$$

Ejemplo:
Consideramos los criptosistemas RSA de claves públicas

$$(328459, 189),\ (328459, 225),\ \text{y}\ (328459, 245).$$

Supongamos que el mensaje $m = 235711$ se envía cifrado con las anteriores claves públicas. Los correspondientes criptogramas son:

$$c_1 = 62705 \qquad c_2 = 157318 \qquad c_3 = 237404$$

Si alguien interceptara estos valores, podría determinar el valor de m. Para ello, resuelve la ecuación:

$$189 \cdot s_1 + 225 \cdot s_2 + 245 \cdot s_3 = 1$$

que tiene solución pues $mcd(189, 225, 245) = 1$.
Vamos a resolver esta ecuación.
Dado que $mcd(189, 225) = 9$, la ecuación $189 \cdot s_1 + 225 \cdot s_2 = 1 - 245 \cdot s_3$ tiene solución si $1 - 245 \cdot s_3$ es múltiplo de 9. Es decir:

$$1 - 245s_3 \equiv 0(\text{mód } 9) \ \Rightarrow\ 7s_3 \equiv 8(\text{mód } 9) \ \Rightarrow\ 28s_3 \equiv 32(\text{mód } 9) \ \Rightarrow\ s_3 \equiv 5(\text{mód } 9)$$

Es decir, $s_3 = 5 + 9 \cdot k$, con k un número entero. Como vamos buscando una solución particular (no todas las soluciones), elegimos $k = 0$, es decir, $s_3 = 5$. En ese caso, la ecuación nos queda:

$$189 \cdot s_1 + 225 \cdot s_2 = 1 - 245 \cdot 5 \ \Rightarrow\ 189 \cdot s_1 + 225 \cdot s_2 = -1224 \ \Rightarrow\ 21 \cdot s_1 + 25 \cdot s_2 = -136$$

Y para resolver esta ecuación calculamos una solución de $21 \cdot s_1' + 25 \cdot s_2' = 1$, que podría ser $s_1' = 6$, $s_2' = -5$. Por tanto, $s_1 = 6 \cdot (-136) = -816$ y $s_2 = (-5) \cdot (-136) = 680$ es una solución de $21 \cdot s_1 + 25 \cdot s_2 = -136$. A partir de esto, la solución general es:

$$\begin{array}{rcrcrcl} s_1 & = & - & 816 & + & 25 \cdot t \\ s_2 & = & & 680 & - & 21 \cdot t \end{array}$$

y para $r = 32$ obtenemos $s_1 = -16$ y $s_2 = 8$, que junto con $s_3 = 5$ nos da la solución que buscábamos.
Si no hubiéramos elegido el valor $k = 0$ para obtener s_3, al final nos habría dado la solución general de la ecuación, que es:

$$\begin{array}{rcrcrcrcl} s_1 & = & - & 816 & - & 1470 \cdot k & + & 25 \cdot t \\ s_2 & = & & 680 & + & 1225 \cdot k & - & 21 \cdot t \\ s_3 & = & & 5 & + & 4 \cdot k & & \end{array}$$

Una vez resuelta la ecuación lo único que tiene que hacer es calcular módulo 328459

$$62705^{-16} \cdot 157318^8 \cdot 237404^5 = (62705^{-1})^{16} \cdot 157318^8 \cdot 237404^5$$

Como $62705^{-1} = 236194$, las cuentas a realizar son:

$$236194^{16} \cdot 157318^8 \cdot 237404^5 = 280315 \cdot 13925 \cdot 328109 = 235711$$

CRIPTOGRAFÍA X

Criptosistema ElGammal

La fortaleza del criptosistema RSA hemos visto que se encuentra en la dificultad de calcular la factorización de un número como producto de números primos (no está demostrado que esto sea así. Hasta ahora las rupturas de un criptosistema RSA implican de una forma u otra la factorización del módulo, pero no hay certeza absoluta de que todas las que existan sean así).
Vamos a describir otro criptosistema que se basa en la dificultad del cálculo del logaritmo discreto.

..... 1

Descripción del algoritmo

Comenzamos explicando cómo obtener las claves:

..... 1.1

Elección de las claves

Para la obtención de las claves seguimos los siguientes pasos:

- Primero elegimos un número primo grande p (de aproximadamente 200 dígitos).
- A continuación, elegimos un elemento α que sea un elemento primitivo de $\mathbb{Z}_p$. Esta condición no es estrictamente necesaria. Si se requiere que α tenga un orden elevado. En general, encontrar un elemento primitivo en $\mathbb{Z}_p$ puede ser bastante complicado. Si elegimos un primo p de forma que $\frac{p-1}{2}$ sea también primo, entonces para comprobar que α es un elemento primitivo basta comprobar que $\alpha \neq p-1$ y $\alpha^{\frac{p-1}{2}} = p-1$ (es decir, que $\left(\frac{\alpha}{p}\right) = -1$ o que α no tiene raíz cuadrada módulo p). Aproximadamente la mitad de los elementos de $\mathbb{Z}_p$ satisfacen tal condición.
- Y por último, elegimos un entero aleatorio x comprendido en el intervalo $1 < x < p-1$. Aunque en teoría con cualquier número en esas condiciones es suficiente, algunos valores de x no son recomendables, como los valores muy pequeños, o muy próximos a $p-1$, o a $\frac{p-1}{2}$, pues en ese caso sería fácil romper el criptosistema.

A partir de estos valores se generan la clave pública y la clave privada.

La clave pública es el trío (p, α, y), donde $y = \alpha^x (\text{mód } p)$.

La clave privada es x.

Ejemplo:
Al igual que con el RSA vamos a dar dos ejemplo: uno con un primo pequeño, y otro con un primo grande.

1. Comenzamos con el primer ejemplo:

   a) Vamos a tomar como primo $p = 533327$. Para este primo, $\frac{p-1}{2} = 266663$ es también primo.

   b) Para buscar un elemento primitivo, vamos probando hasta dar con uno. Como $2^{266663} = 3^{266663} = 1$, ninguno de ellos es primitivo. Pero $5^{266663} = 533326$, luego 5 es un elemento primitivo. Para obtener cualquier otro basta elevar 5 a una potencia que sea prima relativa con $p-1$. Por ejemplo, $5^{125} = 524111$. Elegimos $\alpha = 524111$ (podríamos, sin problema, haber elegido $\alpha = 5$).

   c) Ahora elegimos al azar un número entre 2 y 533325. Por ejemplo, $x = 451225$. Esta será la clave privada.

   d) Calculamos $y = \alpha^x$, y nos da $y = 346395$.

La clave pública es $(533327, 524111, 346395)$, mientras que la clave privada es 451225.
La obtención de la clave privada a partir de la pública requiere la resolución de un problema de logaritmo discreto.

. . . . . 1.2

Cifrado y descifrado

Una vez que se han generado las claves, podemos cifrar un mensaje. El mensaje debe ser un elemento de $\mathbb{Z}_p$, m. Para cifrar:
- Se elige un entero aleatorio k comprendido entre 2 y $p-2$, y que sea primo relativo con $p-1$.
- Se calcula $r = \alpha^k$ (módulo p).
- Se calcula $s = m \cdot y^k$.
- Se envía el par (r, s), que es el mensaje cifrado.

Una vez recibido el mensaje cifrado, el poseedor de la clave privada puede descifrarlo sin más que calcular

$$s \cdot r^{-x} = s \cdot r^{p-1-x}$$

Efectivamente, se tiene que:

$$s \cdot r^{-x} = m \cdot y^k \cdot (\alpha^k)^{-x} = m \cdot (\alpha^x)^k \cdot \alpha^{-x \cdot k} = m$$

Obviamente, todas las operaciones están hechas módulo p.
Una ventaja de este criptosistema es que el mismo mensaje puede ser cifrado de formas diferentes sin más que elegir diferentes valores de k. Presenta, no obstante, el inconveniente de que el tamaño del mensaje cifrado es el doble que el del texto en claro, lo que supone un aumento enlas necesidades de almacenamiento.
Ejemplo:

1. Supongamos que desamos cifrar el mensaje $m = 473210$ con el criptosistema ElGammal descrito en el ejemplo anterior.

   Elegimos un número k, por ejemplo, $k = 273819$.

   Calculamos $r = \alpha^k = 524111^{273819} = 13017$.

   Calculamos $s = m \cdot y^k = 473210 \cdot 346395^{273819} = 473210 \cdot 159487 = 272827$.

   El mensaje cifrafo es por tanto $(13017, 272827)$.

   Para descifrarlo calculamos:

   $$s \cdot r^{p-1-x} = 272827 \cdot 13017^{533326-451225} = 272827 \cdot 13017^{82101} = 272827 \cdot 277881 = 473210$$

   Notemos que para otro valor de k, por ejemplo $k = 192837$, el mensaje cifrado sería $(436906, 389358)$, completamente distinto del anterior. Para el descifrado habría que calcular $389358 \cdot 436906^{82101}$, y el resultado es, como cabría esperar, 473210.

Apéndice: Fundamentos matemáticos

. 1
Aritmética modular
. . . . . 1.1
Generalidades

En esta sección vamos a recordar algunos resultados referentes a la aritmética modular, así como la forma de realizar diversos cálculos:
En primer lugar recordemos como se define dicha aritmética. Para esto, sea n un número natural mayor o igual que 2. En el conjunto de los números enteros definimos la relación:

$$a \equiv b(\text{mód } n) \text{ si } \ b - a \text{ es múltiplo de } n$$

Esta relación es una relación de equivalencia, lo que permite construir el conjunto cociente que se denota por $\mathbb{Z}_n$. A la clase de un elemento $a \in \mathbb{Z}_n$ la denotaremos por $[a]_n$. Es fácil ver que

$$[a]_n = \{\cdots, a-3n, a-2n, a-n, a, a+n, a+2n, a+3n, \cdots\} = a + n\mathbb{Z}$$

Por ejemplo:

$$[0]_2 = \{\cdots, -4, -2, 0, 2, 4, 6, \cdots\} \qquad [1]_2 = \{\cdots, -5, -3, -1, 1, 3, 5, \cdots\}$$

$$[0]_3 = \{\cdots, -6, -3, 0, 3, 6, \cdots\} \qquad [1]_3 = \{\cdots, -5, -2, 1, 4, 7, \cdots\} \qquad [2]_3 = \{\cdots, -4, -1, 2, 5, \cdots\}$$

Puesto que dado cualquier número entero a podemos encontrar c y r tales que
$a = c \cdot n + r$ y $0 \leq r \leq n-1$
entonces para cualquier $a \in \mathbb{Z}$, existe un número r entre 0 y $n-1$ tal que $[a]_n = [r]_n$ (justamente el resto de dividir a entre n).
De esta forma, podemos ver que $\mathbb{Z}_n = \{[0]_n, [1]_n, \cdots, [n-1]_n\}$. Además, es fácil comprobar que estas n clases de equivalencia son todas diferentes, luego $\mathbb{Z}_n$ tiene n elementos.
Para poder definir una aritmética en $\mathbb{Z}_n$ necesitamos los siguientes resultados:

> Sean a, b, c, d números enteros, y n un número natural mayor o igual que 2. Entonces:

$$\left.\begin{array}{l} a \equiv c (\text{mód } n) \\ b \equiv d (\text{mód } n) \end{array}\right\} \implies a + b \equiv b + d (\text{mód } n)$$

$$\left.\begin{array}{l} a \equiv c (\text{mód } n) \\ b \equiv d (\text{mód } n) \end{array}\right\} \implies a \cdot b \equiv b \cdot d (\text{mód } n)$$

A partir de esto podemos definir una suma y una multiplicación en $\mathbb{Z}_n$ como sigue:

$$[a]_n + [b]_n = [a + b]_n \qquad [a]_n \cdot [b]_n = [a \cdot b]_n$$

Es decir, para sumar dos clases, se toman representantes de cada una de ellas, se suman los representantes (que son números enteros) y del resultado se toma su clase. El resultado anterior nos dice que el resultado final no depende de los representantes que elijamos. Para multiplicar se procede de igual forma.
Por ejemplo:

$[3]_7 + [2]_7 = [5]_7$.

$[10]_7 + [-12]_7 = [-2]_7 = [5]_7$. Nótese que $[3]_7 = [10]_7$, mientras que $[2]_7 = [-12]_7$, por tanto, el resultado de la suma debe ser el mismo en ambos casos. Esto es lo que decíamos de que no depende de los representantes elegidos. Para sumar la clase de 3 con la clase de 2, puedo tomar como representante de la clase de 3 el 3, el 10 o cualquier otro, y como representante de $[2]_7$ puedo tomar 2, 9, -12 o cualquier otro. En cualquiera de los casos, el resultado de la suma es $[5]_7$.

$[5]_9 + [7]_9 = [12]_9 = [3]_9$.

$[3]_5 \cdot [4]_5 = [2]_5$.

$[12]_{30} \cdot [20]_{30} = [0]_{30}$.

Si hubiéramos definido la división como $\frac{[a]_n}{[b]_n} = \left[\frac{a}{b}\right]_n$ (siempre que $\frac{a}{b}$ tenga sentido) nos podríamos encontra con cosas como esta:

$$\frac{[8]_{18}}{[4]_{18}} = [2]_{18} \quad \frac{[44]_{18}}{[4]_{18}} = [11]_{18}$$

y vemos que $[8]_{18} = [44]_{18}$, mientras que el resultado de la "división" nos sale diferente. Es decir, la división depende del representante elegido, y por tanto no está bien definida. Para la resta no habría ningún problema, pues la resta se puede poner en función de la suma y el producto: $a - b = a + (-1) \cdot b$.

De ahora en adelante, para representar la clase de un elemento $[a]_n \in \mathbb{Z}_n$ lo haremos simplemente como a. El contexto nos dirá si a es un número entero, o un elemento de algún $\mathbb{Z}_n$.

Hemos visto que en general no tiene mucho sentido hablar de $\frac{a}{b}$ en $\mathbb{Z}_n$. Sin embargo, por ejemplo si tomamos $n = 7$, podemos ver que el único valor coherente para $\frac{2}{3}$ es 3 (ya que 3 es el único elemento de $\mathbb{Z}_7$ que al multiplicarlo por 3 da 2.

$$3 \cdot 0 = 0 \quad 3 \cdot 1 = 3 \quad 3 \cdot 2 = 6 \quad \mathbf{3} \cdot 3 = 2 \quad 3 \cdot 4 = 5 \quad 3 \cdot 5 = 1 \quad 3 \cdot 6 = 4$$

Vemos también que podría tener sentido definir

$$\frac{0}{3} = 0; \quad \frac{1}{3} = 5; \quad \frac{2}{3} = 3; \quad \frac{3}{3} = 1; \quad \frac{4}{3} = 6; \quad \frac{5}{3} = 4; \quad \frac{6}{3} = 2;$$

Por tanto, en $\mathbb{Z}_7$ podría definirse la *división por 3*. Además, definido lo que vale $\frac{1}{3}$ (que es 5), podríamos obtener fácilmente cualquier otra división por 3. Por ejemplo, $\frac{4}{3} = 4 \cdot \frac{1}{3} = 4 \cdot 5 = 6$. Nótese que lo que define al elemento $5 = \frac{1}{3}$ es que $5 \cdot 3 = 1$. En tal caso, diremos que 5 es el inverso (multiplicativo) de 3 módulo 7.

. . . . . 1.2

Cálculo de inversos

Sea $a \in \mathbb{Z}_n$. Se dice que a tiene inverso, que a es invertible, o que a es una unidad si existe $b \in \mathbb{Z}_n$ tal que $b \cdot a = 1$.

Caso de existir ese elemento, puede demostrarse que es único. Este elemento será denotado por a^{-1}

Al conjunto de las unidades de $\mathbb{Z}_n$ lo denotaremos por $\mathcal{U}(\mathbb{Z}_n)$.
Lo que necesitamos ahora es saber cuando un elemento tiene inverso módulo n, y caso de tenerlo, saber calcularlo.

Sea $a \in \mathbb{Z}_n$. Entonces a es una unidad si, y sólo si, $mcd(a, n) = 1$.

Por ejemplo, puesto que $mcd(3, 7) = 1$ se tiene que 3 es una unidad en $\mathbb{Z}_7$. Por otra parte, como $mcd(4, 18) \neq 1$ entonces 4 no tiene inverso módulo 18 (éste es el motivo de que no podamos dividir por 4 en $\mathbb{Z}_{18}$).
Para calcular el inverso de un elemento en $\mathbb{Z}_n$, podemos ir multiplicándolo por los distintos elementos de $\mathbb{Z}_n$ hasta que nos de 1. En $\mathbb{Z}_{12}$ el 5 tiene inverso. Para hallarlo:

$$5 \cdot 1 = 5 \qquad 5 \cdot 2 = 10 \qquad 5 \cdot 3 = 3 \qquad 5 \cdot 4 = 8 \qquad 5 \cdot 5 = 1$$

luego $5^{-1} = 5$.
Sin embargo, este método es muy poco eficiente para números grandes. Ahora bien, sabemos que para calcular el máximo común divisor podemos emplear el algoritmo de Euclides. Este algoritmo consiste en ir realizando divisiones sucesivas hasta obtener resto 0. El último resto distinto de cero es el máximo común divisor de los dos números. Vamos a verlo con un ejemplo:
Queremos calcular el máximo común divisor de 391 y 1542. Procedemos como sigue:

$1542 = 3 \cdot 391 + 369$
$391 = 1 \cdot 369 + 22$
$369 = 16 \cdot 22 + 17$
$22 = 1 \cdot 17 + 5$
$17 = 3 \cdot 5 + 2$
$5 = 2 \cdot 2 + 1$
Luego $\mathrm{mcd}(1542, 391) = 1$. Por tanto, 391 tiene inverso en $\mathbb{Z}_{1542}$. Vamos a calcularlo:

$1542 = 3 \cdot 391 + 369$	369	$= 1 \cdot 1542 + (-3) \cdot 391$
$391 = 1 \cdot 369 + 22$	$22 = 391 - 369$	$= 391 - (1 \cdot 1542 + (-3) \cdot 391)$ $= (-1) \cdot 1542 + 4 \cdot 391$
$369 = 16 \cdot 22 + 17$	$17 = 369 - 16 \cdot 22$	$= 1 \cdot 1542 + (-3) \cdot 391 - 16 \cdot ((-1) \cdot 1542 + 4 \cdot 391)$ $= 17 \cdot 1542 + (-67) \cdot 391$
$22 = 1 \cdot 17 + 5$	$5 = 22 - 17$	$= ((-1) \cdot 1542 + 4 \cdot 391) - (17 \cdot 1542 + (-67) \cdot 391)$ $= (-18) \cdot 1542 + 71 \cdot 391$
$17 = 3 \cdot 5 + 2$	$2 = 17 - 3 \cdot 5$	$= (17 \cdot 1542 + (-67) \cdot 391) - 3 \cdot ((-18) \cdot 1542 + 71 \cdot 369)$ $= 71 \cdot 1542 + (-280) \cdot 391$
$5 = 2 \cdot 2 + 1$	$1 = 5 - 2 \cdot 2$	$= ((-18) \cdot 1542 + 71 \cdot 369) - 2 \cdot (71 \cdot 1542 + (-280) \cdot 369)$ $= (-160) \cdot 1542 + 631 \cdot 391$

Si en la igualdad $1 = (-160) \cdot 1542 + 631 \cdot 391$ reducimos módulo 1542 nos queda que $1 = (-160) \cdot 0 + 631 \cdot 391 = 631 \cdot 391$, luego el inverso de 391 en $\mathbb{Z}_{1542} = 631$.
Vamos a ver el camino seguido para obtener este número. Para eso nos fijamos en lo que en cada etapa va multiplicando a 391. Vamos a denotar este número por v.

$v = -3$
$v = 1 - \cdot(-3) = 4$
$v = -3 - 16 \cdot 4 = -67$
$v = 4 - 1 \cdot (-67) = 71$
$v = -67 - 3 \cdot 71 = -280$
$v = 71 - 2 \cdot (-280) = 631$

El siguiente algoritmo recoge esta idea para el cálculo de inversos modulares:

Algoritmo INVERSO(n, a)

Entrada: $n, a \in \mathbb{Z} : n \geq 2$

Salida: u: $u = a^{-1}$ en $\mathbb{Z}_n$ (si existe)

$(y, v) := (0, 1)$

$r := n$ mód a

Mientras $r \neq 0$
    $c := n$ div a
    $(y, v) := (v, y - v \cdot c)$
    $(n, a) := (a, r)$
    $r := n$ mód a
Si $a \neq 1$
    Devuelve "No existe inverso"
    Fin
Devuelve v
Fin

En el ejemplo que acabamos de ver nos quedaría:

n	a	r	c	y	v
				0	1
1542	391	369	3	1	-3
391	369	22	1	-3	4
369	22	17	16	4	-67
22	17	5	1	-67	71
17	5	2	3	71	-280
5	2	1	2	-280	631

Nótese que si p es un número primo, entonces todo elemento de $\mathbb{Z}_p$, salvo el cero, tiene inverso.
Vamos a denotar por $\varphi(n)$ al número de unidades de $\mathbb{Z}_n$, es decir, el número de elementos menores que n que son primos relativos con n. Esta función se denomina *función de Euler*.
Si p es un número primo, entonces $\varphi(p) = p - 1$ (pues todos los elementos salvo el cero son unidades). Más general, si p es un número primo, entonces $\varphi(p^e) = p^e - p^{e-1} = p^e \cdot \left(1 - \frac{1}{p}\right)$.
Las unidades de $\mathbb{Z}_{p^e}$ son todos los elementos de $\mathbb{Z}_{p^e}$ menos los que son múltiplos de p, y éstos son $0 \cdot p, 1 \cdot p, 2 \cdot p, \cdots, (p^{e-1} - 2) \cdot p, (p^{e-1} - 1) \cdot p$.
Por ejemplo, el conjunto de las unidades de $\mathbb{Z}_{27}$ es:

$$\{1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26\}$$

que podemos ver que tiene $27-9 = 18$ elementos (de $\mathbb{Z}_{27}$ se han quitado $0, 3, 6, 9, 12, 15, 18, 21, 24$).
Si m y n son primos relativos, entonces $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$. Por ejemplo, $\varphi(15) = \varphi(3) \cdot \varphi(5) = 2 \cdot 4 = 8$. El conjunto de elementos invertibles de $\mathbb{Z}_{15}$ es $\{1, 2, 4, 7, 8, 11, 13, 14\}$, que tiene 8 elementos.
Si m y n no son primos relativos no es cierto que $\varphi(m \cdot m) = \varphi(m) \cdot \varphi(n)$. Por ejemplo, $\varphi(6) = 2$, $\varphi(9) = 6$, mientras que $\varphi(54) = 18 \neq 2 \cdot 6$

$$\mathcal{U}(\mathbb{Z}_6) = \{1, 5\}; \qquad \mathcal{U}(\mathbb{Z}_9) = \{1, 2, 4, 5, 7, 8\};$$

$$\mathcal{U}(\mathbb{Z}_{54}) = \{1, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53\}.$$

Para calcular $\varphi(n)$ lo que podemos hacer es descomponer n como producto de potencias de primos, y calcular la función de Euler de cada uno de los factores que nos han aparecido. Después se multiplican todos los resultados.
Es decir, si $n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_r^{e_r}$ entonces:

$$\begin{aligned}\varphi(n) = \varphi(p_1^{e_1}) \cdot \varphi(p_2^{e_2}) \cdots \varphi(p_r^{e_r}) &= (p_1^{e_1} - p_1^{e_1-1}) \cdot (p_2^{e_2} - p_2^{e_2-1}) \cdots (p_r^{e_r} - p_r^{e_r-1}) \\ &= n \cdot \left(1 - \tfrac{1}{p_1}\right) \cdot \left(1 - \tfrac{1}{p_2}\right) \cdots \left(1 - \tfrac{1}{p_r}\right)\end{aligned}$$

. . . . . 1.3

Cálculo de potencias

Vamos a ver aquí cómo calcular potencias modulares, aunque lo que digamos vale también para cualquier otro tipo de potencias.
El problema es, dados a, m y n calcular a^m (mód n). Lo primero que se nos ocurre es multiplicar a consigo mismo m veces, pero esto es muy ineficiente (es como si para multiplicar un número por un número natural realizaramos tantas sumas como nos indica el número natural por el que multiplicamos). Un primer resultado que nos puede simplificar en ocasiones los cálculos es el siguiente:

> **Teorema de Fermat** Dados $a, n \in \mathbb{N}$, si $mcd(a, n) = 1$ entonces $a^{\varphi(n)} \equiv 1 (\text{mód } n)$.

Así, por ejemplo, si queremos calcular 7^{111} en $\mathbb{Z}_{53}$, puesto que $mcd(7, 53) = 1$ entonces $7^{52} = 1$, luego $7^{104} = 1$, y por tanto $7^{111} = 7^7 = 29$.
Para calcular 7^{111} en $\mathbb{Z}_{54}$, vemos en primer lugar que $mcd(7, 54) = 1$. Calculamos entonces $\varphi(54)$. Como $54 = 2 \cdot 27$ entonces $\varphi(54) = \varphi(2) \cdot \varphi(27) = 18$. Por tanto, $7^{18} = 1$.
Puesto que $111 = 18 \cdot 6 + 3$ tenemos que:

$$7^{111} = 7^{18 \cdot 6 + 3} = 7^{18 \cdot 6} \cdot 7^3 = (7^{18})^6 \cdot 7^3 = 1^6 \cdot 7^3 = 7^3 = 19$$

Pero, ¿qué ocurre si $mcd(a, n) \neq 1$?, ¿o cuando, aún siendo a y n primos relativos, el exponente al que hay que elevar a es muy elevado?
Vamos a calcular 1234^{3568} en $\mathbb{Z}_{21367}$. Sería un error intentar calcular 1234^{3568} en $\mathbb{Z}$ y el resultado reducirlo módulo 21357, ya que el número 1234^{3568} tiene más de 11000 cifras (en concreto tiene 11030).
Veamos una forma de calcularlo. Para ello procedemos como sigue:

$1234^1 = 1234.$

$1234^2 = 1522756 = 5699.$

$1234^4 = 5699^2 = 32478601 = 761.$

$1234^8 = 761^2 = 579121 = 2212.$

$1234^{16} = 2212^2 = 4892944 = 21268.$

$1234^{32} = 21268^2 = 452327824 = 9801.$

$1234^{64} = 9801^2 = 96059601 = 14936.$

$1234^{128} = 14936^2 = 223084096 = 12616.$

$1234^{256} = 12616^2 = 159163456 = 673.$

$1234^{512} = 673^3 = 452929 = 4222.$

$1234^{1024} = 4222^2 = 17825284 = 5206.$

$1234^{2048} = 5206^2 = 27102436 = 9080.$

Ahora, como $3568 = 2048 + 1024 + 256 + 128 + 64 + 32 + 16$, tenemos que

$$7^{3568} = 7^{2048} \cdot 7^{1024} \cdot 7^{256} \cdot 7^{128} \cdot 7^{64} \cdot 7^{32} \cdot 7^{16}.$$

Luego:

$7^{32} \cdot 7^{16} = 9801 \cdot 21268 = 208447668 = 12583.$

$7^{64} \cdot 7^{32} \cdot 7^{16} = 14936 \cdot 12583 = 187939688 = 16923.$

$7^{128} \cdot 7^{64} \cdot 7^{32} \cdot 7^{16} = 12616 \cdot 16923 = 213500568 = 1504.$

$7^{256} \cdot 7^{128} \cdot 7^{64} \cdot 7^{32} \cdot 7^{16} = 673 \cdot 1504 = 1012192 = 7943.$

$7^{1024} \cdot 7^{256} \cdot 7^{128} \cdot 7^{64} \cdot 7^{32} \cdot 7^{16} = 5206 \cdot 7943 = 41351258 = 6113.$

$7^{3568} = 7^{2048} \cdot 7^{1024} \cdot 7^{256} \cdot 7^{128} \cdot 7^{64} \cdot 7^{32} \cdot 7^{16} = 9080 \cdot 6113 = 55506040 = 15941.$

Notemos que para calcular a^m lo que hacemos es calcular a elevado a las potencias de 2, y luego elegimos cuáles de ellas hemos de multiplicar. Éstas las conseguimos expresando el exponente como suma de potencias de 2, o lo que es lo mismo, calculando la expresión binaria del exponente. En nuestro caso se tiene que $3568 = 110111110000)_2$. Esta expresión la podemos obtener realizando divisiones del exponente por 2 y tomando los restos.
El siguiente algoritmo nos calcula a^m (mód n).

Algoritmo POTENCIA(a, m, n)
Entrada: $a, m, n \in \mathbb{Z} : m \geq 1; n \geq 2$
Salida: b: $u = a^m$ en $\mathbb{Z}_n$
    $b := 1$
    Mientras $m \neq 1$

Si m mód $2 = 1$
$b := b \cdot a$
$a := a^2$
$m := m$ div 2
$b := a \cdot b$
Devuelve b
Fin

b	a	m	m mód 2
1	1234	3568	0
	5699	1784	0
	761	892	0
	2212	446	0
	21268	223	1
21268	9801	111	1
12583	14936	55	1
16923	12616	27	1
1504	673	13	1
7943	4222	6	0
	5206	3	1
6113	9080	1	
15941			

Los tres primeros valores son los valores iniciale. A continuación se rellena la tabla de izquierda a derecha, y de arriba hacia abajo. La última columna es sólo para comprobar. Cuando vale cero, en la fila siguiente no se modifica b.

> **Nota:** Al igual que este algoritmo que realiza muchas multiplicaciones por un mismo número, podríamos dar otro que realizara sumas por el mismo número (es decir, multiplicaría). En este caso, el cálculo de cuadrados sería multiplicación por 2 (que a nivel de bits sería añadir un cero a la derecha).

Como consecuencia de un resultado de teoría de grupos (teorema de Lagrange), y que las unidades de $\mathbb{Z}_n$ forman un grupo, tenemos el siguiente resultado:

Teorema 1.1. Teorema pequeño de Fermat
Sea p un número primo, y a un número entero tal que $1 \leq a \leq p-1$. Entonces

$$a^{p-1} \equiv 1 (\text{mód } p)$$

Este teorema es un caso particular del teorema de Fermat que hemos visto antes.
Nótese que si tomamos un número n, y calculamos $a^{n-1}(\text{mód } n)$ para algún número entero a tal que $1 \leq a \leq n-1$, y nos sale distinto de 1, entonces el número n no puede ser primo.

Ejemplo 1.1. Tomamos $n = 21$, y $a = 2$. Entonces $a^{20} = 1048576$, luego $2^{20} \equiv 4(\text{mód } 21)$. Por tanto, podemos concluir que 21 no es primo.
Esto no se puede tomar al revés. Por ejemplo, 15 no es primo, y sin embargo $4^{14} = 268435456$, que es congruente con 1 módulo 15 (sin embargo, $2^{14} = 16384$, que no es congruente con 1 módulo 15).

Más adelante, cuando veamos los test de primalidad profundizaremos en esta idea.

. . . . . 1.4

Raíces cuadradas modulares

Vamos a ver aquí como calcular raíces cuadradas módulo un primo. Es decir, vamos a ver cuántas soluciones, entre 0 y $p-1$, tiene la congruencia $x^2 \equiv a(\text{mód } p)$.
Dicho de otra forma vamos a calcular en $\mathbb{Z}_p$ las raíces del polinomio $x^2 - a$. Al ser p primo, el número máximo de raíces viene dado por el grado del polinomio, es decir, 2. Por tanto el número de raíces puede ser 0, 1 ó 2.
Sabemos que si r es una raíz de $x^2 - a$, entonces $-r$ también lo es. Por tanto, para que $x^2 - a$ tenga sólo una raíz r es necesario que $r = -r$. Esto ocurre únicamente si $r = 0$ (salvo que el primo p sea 2). En principio vamos a interesarnos por saber si $x^2 - a$ tiene o no raíces en $\mathbb{Z}_p$.

Definición 12. Sea p un número primo impar, y a un número entero. Se dice que a es un residuo cuadrático módulo p si $x^2 - a$ tiene dos raíces en $\mathbb{Z}_p$
A partir de esto se define el *símbolo de Legendre* de a y p como sigue:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{si } p|a \\ 1 & \text{si } a \text{ es un residuo cuadrático módulo } p \\ -1 & \text{si } x^2 - a \text{ no tiene raíces en } \mathbb{Z}_p \end{cases}$$

Hasta ahora, únicamente le hemos asignado un número al hecho de tener o no tener raíz cuadrada módulo p. A continuación veremos cómo calcular el símbolo de Legendre.
Algunas propiedades elementales son:

- $\left(\frac{a}{p}\right) = \left(\frac{a \text{ mód } p}{p}\right)$.
- $\left(\frac{a \cdot b}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$.
- $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}}(\text{mód } p)$.
- $\left(\frac{a^2}{p}\right) = 1$ si $\text{mcd}(a, p) = 1$.

La primera propiedad es evidente a partir de la definición, así como la cuarta. La segunda nos dice que el producto de dos elementos tiene raíz cuadrada módulo p si, y sólo si, los dos tienen raíz cuadrada o ninguno de ellos la tiene.
Otras propiedades útiles para el cálculo del símbolo de Legendre son:

1. $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1 & \text{si } p \equiv 1(\text{mód } 4) \\ -1 & \text{si } p \equiv 3(\text{mód } 4) \end{cases}$

2. $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1 & \text{si } p \equiv \pm 1 (\text{mód } 8) \\ -1 & \text{si } p \equiv \pm 3 (\text{mód } 8) \end{cases}$

3. Si p y q son primos impares entonces

$$\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)\cdot(q-1)}{4}} \left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{p}{q}\right) & \text{si } p \equiv q \equiv 3 (\text{mód } 4) \\ \left(\frac{p}{q}\right) & \text{en otro caso.} \end{cases}$$

Este último resultado se conoce como *Ley de reciprocidad cuadrática*. Con estas últimas propiedades podemos calcular el símbolo de Legendre de dos números (el segundo primo). Por ejemplo:

$\left(\frac{646}{809}\right) = \left(\frac{2}{809}\right) \cdot \left(\frac{323}{809}\right)$

Como $809 \equiv 1 (\text{mód } 8)$ se tiene que $\left(\frac{2}{809}\right) = 1$, y como $323 = 17 \cdot 19$ entonces $\left(\frac{323}{809}\right) = \left(\frac{17}{809}\right) \cdot 19809 = \left(\frac{809}{17}\right) \cdot \left(\frac{809}{19}\right)$, ya que $809 \equiv 1 (\text{mód } 4)$. Por tanto

$\left(\frac{646}{809}\right) = 1 \cdot \left(\frac{809}{17}\right) \cdot \left(\frac{809}{19}\right) = \left(\frac{10}{17}\right) \cdot \left(\frac{11}{19}\right) = \left(\frac{2}{17}\right) \cdot \left(\frac{5}{17}\right) \cdot \left(\frac{11}{19}\right)$.

Puesto que $17 \equiv 1 (\text{mód } 8)$, por la propiedad segunda se tiene que $\left(\frac{2}{17}\right) = 1$.

$\left(\frac{5}{17}\right) = \left(\frac{17}{5}\right)$, ya que $5 \equiv 1 (\text{mód } 4)$, y $\left(\frac{17}{5}\right) = \left(\frac{2}{5}\right) = -1$ puesto que $5 \equiv -3 (\text{mód } 8)$.

$\left(\frac{11}{19}\right) = -\left(\frac{19}{11}\right) = -\left(\frac{8}{11}\right) = -\left(\frac{2}{11}\right)^3 = -(-1)^3 = 1$.

$\left(\frac{646}{809}\right) = 1 \cdot (-1) \cdot 1 = -1$.

Por tanto, no existe raíz cuadrada de 558 en $\mathbb{Z}_{809}$.

A la hora de calcular el símbolo de Legendre vemos que probablemente tengamos que realizar una factorización de un número como producto de primos. Esto podría ralentizar mucho el cálculo. Para evitar esto, se extiende el símbolo de Legendre al caso en que el segundo número sea impar, sin necesidad de que sea primo. Surge así lo que se conoce como *símbolo de Jacobi*

Definición 13. Sea a un número entero, y n un número impar. Supongamos que la factorización de n como producto de primos es $n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$. Se define el símbolo de Jacobi de a y n como:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \cdot \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

El símbolo de Jacobi no tiene ningún significado referente a la existencia de raíces cuadradas. Por ejemplo:

$$\left(\frac{5}{299}\right) = \left(\frac{5}{13}\right) \cdot \left(\frac{5}{23}\right) = \left(\frac{13}{5}\right) \cdot \left(\frac{23}{5}\right) = \left(\frac{3}{5}\right) \cdot \left(\frac{3}{5}\right) = 1$$

y sin embargo, no existe raíz cuadrada de 5 en $\mathbb{Z}_{299}$.

El símbolo de Jacobi es una herramienta para el cálculo del símbolo de Legendre. Y es que las propiedades vistas para el símbolo de Legendre se extienden ahora al símbolo de Jacobi. Las enumeramos a continuación.
Sean a, b números enteros, y n un número impar. Entonces:

1. $\left(\frac{a}{n}\right) = \left(\frac{a \bmod n}{n}\right)$.
2. $\left(\frac{a \cdot b}{n}\right) = \left(\frac{a}{n}\right) \cdot \left(\frac{b}{n}\right)$.
3. $\left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} (\text{mód } n)$.
4. $\left(\frac{a^2}{n}\right) = 1$ si $mcd(a, n) = 1$.
5. $\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}} = \begin{cases} 1 & \text{si } n \equiv 1 (\text{mód } 4) \\ -1 & \text{si } n \equiv 3 (\text{mód } 4) \end{cases}$
6. $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}} = \begin{cases} 1 & \text{si } n \equiv \pm 1 (\text{mód } 8) \\ -1 & \text{si } n \equiv \pm 3 (\text{mód } 8) \end{cases}$
7. Si n y m son números impares entonces
$$\left(\frac{m}{n}\right) = (-1)^{\frac{(n-1)\cdot(m-1)}{4}} \left(\frac{n}{m}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{si } n \equiv m \equiv 3 (\text{mód } 4) \\ \left(\frac{n}{m}\right) & \text{en otro caso.} \end{cases}$$

Veamos los dos ejemplos que acabamos de hacer:
$\left(\frac{5}{299}\right) = \left(\frac{299}{5}\right) = \left(\frac{4}{5}\right) = \left(\frac{2}{5}\right)^2 = 1$
Aunque como hemos visto, esto no tiene ningún significado.

$$\begin{aligned} \left(\frac{646}{809}\right) &= \left(\frac{2}{809}\right) \cdot \left(\frac{323}{809}\right) = 1 \cdot \left(\frac{809}{323}\right) = \left(\frac{163}{323}\right) = -\left(\frac{323}{163}\right) = -\left(\frac{160}{163}\right) \\ &= -\left(\frac{2^5}{163}\right) \cdot \left(\frac{5}{163}\right) = -(-1)^5 \cdot \left(\frac{163}{5}\right) = \left(\frac{3}{5}\right) = \left(\frac{5}{3}\right) = \left(\frac{2}{3}\right) = -1 \end{aligned}$$

Ya sabemos, dado un primo p y un número a, si a tiene o no raíz cuadrada módulo p. Lo que pretendemos ahora es, sabiendo que a tiene raíz cuadrada, calcularla.
Supongamos que tenemos un número primo p, un número a tal que $\left(\frac{a}{p}\right) = 1$. Procedemos entonces como sigue:

- Buscamos n tal que $\left(\frac{n}{p}\right) = -1$.
- Descomponemos $p - 1$ como $p - 1 = 2^u \cdot s$, con s un número impar.
- Si $u = 1$ hacemos $r = a^{\frac{p+1}{4}}$.
- Si $u \geq 2$ entonces:
  - Hacemos $r = a^{\frac{s+1}{2}}$.
  - Hacemos $b = n^s$.
  - Hacemos $j = 0$.
  - Mientras $j \leq u - 2$.

- Si $(a^{-1}r^2)^{2^{u-2-j}} = -1$ hacemos $r = r \cdot b$.
- Hacemos $b = b^2$.
- Hacemos $j = j + 1$.

- Devuelve r.

Algunas observaciones:
En $\mathbb{Z}_p$, la mitad de los elementos no son residuos cuadráticos. Por tanto, una forma de encontrar uno es calculando $\left(\frac{2}{p}\right)$, $\left(\frac{3}{p}\right)$, etc, hasta que nos de -1.
Para descomponer $p-1$ como $2^u \cdot s$ basta ir dividiendo $p-1$ por 2 hasta que obtengamos un número impar. Este número es s, y el número divisiones u.
La condición *Si* $u = 1$ no es necesaria. u toma el valor 1 cuando $p \equiv 3(\text{mód } 4)$. En este caso, $s = \frac{p-1}{2}$, luego $\frac{s+1}{2} = \frac{\frac{p-1}{2}+1}{2} = \frac{p+1}{4}$, luego al calcular $a^{\frac{s+1}{2}}$ calcularíamos $a^{\frac{p+1}{4}}$. Además, si $u = 1$ no entraría en el bucle.
Las potencias (salvo 2^{u-2-j}) y las multiplicaciones, así como el cálculo del inverso hay que realizarlos módulo p, y cuando pregunta $(a^{-1}r^2)^{2^{u-2-j}} = -1$ también es módulo p, luego podemos sustituirla por $(a^{-1}r^2)^{2^{u-2-j}} = p-1$.
Seguidamente vamos a justificar porqué el anterior algoritmo nos devuelve una raíz cuadrada de a.

- El algoritmo, lo que hace es buscar un elemento r tal que $a^{-1} \cdot r^2 = 1$. Dicho elemento r será la raíz cuadrada.

- Sobre el elemento $b = n^s$ sabemos que es una raíz de índice 2^u de la unidad y primitiva, pues:

  $$b^{2^u} = (n^s)^{2^u} = n^{s \cdot 2^u} = n^{p-1} = 1$$

  Esto nos dice que es una raíz de índice 2^u. Vamos a ver que es primitiva, es decir, que el primer exponente al que hay que elevar b para que dé 1 es 2^u. Este primer exponente debe ser un divisor de 2^u, y como los únicos divisores de 2^u son potencias de 2, basta comprobar que $b^{2^{u-1}} \neq 1$.

  $$b^{2^{u-1}} = (n^s)^{2^{u-1}} = n^{s \cdot 2^{u-1}} = n^{\frac{p-1}{2}} = \left(\frac{n}{p}\right) = -1$$

  Como consecuencia de esto, el conjunto

  $$\{1, b, b^2, \cdots, b^{2^u-2}, b^{2^u-1}\}$$

  tiene exactamente 2^u elementos, y éstos son las raíces del polinomio $x^{2^u} - 1$, y por tanto, los elementos del conjunto

  $$\{1, b^2, b^4, \cdots, b^{2^u-4}, b^{2^u-2}\}$$

son las raíces de índice 2^{u-1} de la unidad. Puesto que $b^{-k} = b^{2^u-k}$ el conjunto anterior es igual a

$$\{1, b^{-2}, b^{-4}, \cdots, b^{-(2^u-4)}, b^{-(2^u-2)}\}$$

- Puesto que $a^{-1} \cdot r^2$ es una raíz de índice 2^{u-1} de la unidad, debe ser uno de los elementos del anterior conjunto, es decir, existe un t entre 0 y $2^{u-1} - 1$ tal que

$$a^{-1} \cdot r^2 = b^{-2t}$$

En tal caso, $1 = a^{-1} \cdot r^2 \cdot b^{2t} = a^{-1} \cdot (r \cdot b^t)^2$, luego $r \cdot b^t$ será la raíz cuadrada de a.

Lo que hacemos es ir modificando r de forma que si en un paso (iteración del bucle) $a^{-1} \cdot r^2$ es una raíz de índice 2^k de la unidad, en la siguiente sea una raíz de índice 2^{k-1} de la unidad.

- Si $(a^{-1} \cdot r^2)^{2^k} = 1$, entonces $(a^{-1} \cdot r^2)^{2^{k-1}}$ es una raíz cuadrada de 1, luego vale 1 o -1. Si vale 1, entonces $a^{-1} \cdot r^2$ es ya una raíz cuadrada de índice 2^{k-1}, por lo que r se queda como está. Pero si vale -1 entonces hemos de modificar r, multiplicándolo por el valor que toma b en ese momento.

Ejemplo 1.2. Vamos a calcular la raíz cuadrada de 319 módulo 353. En primer lugar comprobamos si dicha raíz existe (bueno, habría que ver antes que 353 es primo).

$$\left(\frac{319}{353}\right) = \left(\frac{353}{319}\right) = \left(\frac{34}{319}\right) = \left(\frac{2}{319}\right)\cdot\left(\frac{17}{319}\right) = \left(\frac{17}{319}\right) = \left(\frac{319}{17}\right) = \left(\frac{13}{17}\right) = \left(\frac{17}{13}\right) = \left(\frac{4}{17}\right) = 1$$

Una vez hecho esto, necesitamos expresar 352 como $2^u \cdot s$, y encontrar n tal que $\left(\frac{n}{353}\right) = -1$. Para lo primero vamos dividiendo 352 por dos hasta que nos dé impar

$$352 \to 176 \to 88 \to 44 \to 22 \to 11$$

es decir, $352 = 2^5 \cdot 11$ ($u = 5;\ s = 11$).
Para encontrar n, vamos probando con 2, 3, etc. hasta que obtengamos uno.

- $\left(\frac{2}{353}\right) = 1$, pues $353 \equiv 1 (\text{mód } 8)$.
- $\left(\frac{3}{353}\right) = \left(\frac{353}{3}\right) = \left(\frac{2}{353}\right) = -1$.

Por tanto, $n = 3$.
Inicializamos las constantes:
$r = a^{\frac{s+1}{2}} = 319^6 = 168$; $b = n^s = 3^{11} = 294$.
Y calculamos a^{-1}, que vale 218.

j	$a^{-1}\cdot r^2$	$(a^{-1}\cdot r^2)^{2^{3-j}}$	r	b
			168	294
0	42	1		304
1	42	1		283
2	42	-1	242	311
3	1	1		352

El valor de r, es decir, 242 es una raíz cuadrada de 319. La otra es $353-242=111$.
Algunas observaciones:

- Podemos crear una variable auxiliar c en la que vamos guardando el valor $a^{-1}\cdot r^2$. Antes de entrar en el bucle la inicializamos, y cada vez que se modifique r (cuando hacemos $r=r\cdot b$) podemos añadir $c=c\cdot b^2$ y actualizamos el valor de c. En ese caso, el condicional quedaría

  Si $c^{2^{u-2-j}}=-1$ hacemos $r=r\cdot b$ y $c=c\cdot b^2$.

- Una vez terminado el bucle podemos añadirle una instrucción para que nos devuelva siempre la menor raíz cuadrada. Esta sería:

  Si $r>\frac{p}{2}$ hacemos $r=p-r$.

  O si preferimos, podemos poner al final Devuelve r, $p-r$.

Nótese que $b^{32}=1$, y 32 es el primer exponente (aparte del cero) para el que ocurre eso. De hecho, las distintas potencias de b son

1	294	304	67	283	247	253	252	311	7	293	10	116	216	317	6
352	59	49	286	70	106	100	101	42	346	60	343	237	137	36	347

Éstas son las 32 raíces trigésimo segundas de la unidad. De ellas, la mitad son raíces décimosextas de la unidas. Éstas son:

1	304	283	253	311	293	116	317	352	49	70	100	42	60	237	36

Una de ellas, la décimotercera de la lista, coincide con $a^{-1}\cdot r^2$. Por tanto, $a^{-1}\cdot r^2=b^{24}=b^{-8}$, luego $1=a^{-1}\cdot r^2\cdot b^8=a^{-1}\cdot(r\cdot b^4)^2$.
Por tanto, el valor inicial de r hay que multiplicarlo por b^4. Dado que la expresión de 4 en binario es 0100, tan solo en la tercera iteración del bucle se altera el valor de r.

. . . . . 1.5

Logaritmo discreto

Hemos visto hace poco, como dados tres números naturales a, b, m, con $m\geq 2$ podemos calcular de forma rápida el valor de $a^b\equiv(\text{mód } m)$. Ahora nos planteamos un problema que podría ser catalogado como el problema inverso. Tenemos a, c, m en las mismas condiciones

que antes y tratamos de encontrar b tal que $a^b \equiv c(\text{mód } m)$, o lo que es lo mismo, $a^b = c$ en $\mathbb{Z}_n$. Este problema se denomina el *problema del logaritmo discreto*.
La primera forma que se ocurre para atacar el problema es por tentativa. Vamos probando con diferentes exponentes, hasta que demos con la solución, si es que tiene. El número de exponentes a probar está acotado superiormente por $\varphi(m)$. Lo cual hace muy grande el número de comprobaciones, incluso para números no excesivamente grandes (por ejemplo, si hiciéramos un millón de comprobaciones por segundo, para un primo del orden de un billón podríamos tardar unos 11 días. Si el primo fuera del orden de mil billones, el tiempo necesario podría subir a más de 30 años).
Vamos a dar un algoritmo, debido a Daniel Shanks, conocido como el algoritmo *paso enano, paso de gigante*. Este algoritmo sirve únicamente si el número m es primo. Por tanto, a partir de ahora nos restringiremos al caso de módulos primos.

Paso enano - Paso gigante

Supongamos que p es un número primo, y tenemos a, c dos números enteros no nulos. Pretendemos encontrar un número b tal que $a^b = c$ en $\mathbb{Z}_p$.
Notemos que por el teorema de Fermat, $a^{p-1} = 1$, luego el exponente b, de existir, podemos encontrarlo en el rango $1 \leq b \leq p-1$.
Antes de continuar, vamos a dar un lema muy sencillo.

> Sea s un número natural. Entonces cualquier número natural n comprendido entre 1 y s^2 se puede expresar de forma única como $n = t \cdot s - r$, con $0 \leq r \leq s-1$ y $1 \leq t \leq s$.
>
> Para comprobarlo, distinguimos dos casos:
>
> - $1 \leq n \leq s^2$, y n no es múltiplo de s. En tal caso, podemos dividir n entre s obteniendo un cociente c y un resto r', tales que $n = c \cdot s + r'$, y $0 \leq c \leq s-1$ (nótese que $n \neq s^2$) y $1 \leq r' \leq s-1$. Basta tomar entonces $t = c+1$ y $r = s - r'$, y se verifica:
>
> $$n = c \cdot s + r' = c \cdot s + s - s + r' = (c+1) \cdot s - (s - r') = t \cdot s - r$$
>
> - n es múltiplo de s. En tal caso, $n = t \cdot s = t \cdot s - 0$, y $1 \leq t \leq s$.
>
> La unicidad se puede demostrar de forma análoga a como se prueba la unicidad el cociente y el resto de una división entera.

La idea del algoritmo es expresar el exponente b de la forma $t \cdot s - r$ para algún número s. Para asegurarnos que todos los posibles exponentes los podemos expresar así necesitamos que $s^2 \geq p-1$.
Por tanto, lo primero que hay que hacer es encontrar un entero s que sea mayor o igual que $\sqrt{p}$. Cuando menor sea s (con esa condición) más rápido será el algoritmo. Una forma de estimar s puede ser usando el método de Newton-Raphson.

Notemos que si $a^b = c$, entonces $a^{t\cdot s} \cdot a^{-r} = b$, es decir, $(a^s)^t = a^r \cdot c$. El algoritmo, lo que hace es constriur dos tablas

$$c;\ a\cdot c;\ a^2\cdot c;\ \cdots\ a^r\cdot c;\ \cdots\ a^{s-1}\cdot c. \qquad a^s;\ a^{2s};\ \cdots\ a^{t\cdot s};\ \cdots\ a^{s\cdot s}.$$

y encontrar coincidencias entre ambas. Si encontramos que $a^r \cdot c = a^{t\cdot s}$, entonces $a^{t\cdot s-r} = c$. Caso de no haber coincidencias, entonces no existe el logaritmo.
Al crear las tablas, hemos de guardar los elementos con el exponente que estamos empleando. Así, tendríamos las dos tablas:

$S =$

c	$a\cdot c$	$a^2\cdot c$	$\cdots$	$a^r\cdot c$	$\cdots$	$a^{s-1}\cdot c$
0	1	2	$\cdots$	r	$\cdots$	$s-1$

$T =$

a^s	a^{2s}	a^{3s}	$\cdots$	$a^{t\cdot s}$	$\cdots$	a^{s^2}
1	2	3	$\cdots$	t	$\cdots$	s

Para obtener el elemento $a^r\cdot c$ de la primera tabla no es necesario realizar el cálculo a^r, y luego multiplicarlo por c, sino que podemos aprovechar el valor que hemos obtenido previamente $(a^{r-1}\cdot c)$, y multiplicarlo por a.
Para obtener el elemento $a^{t\cdot s}$ de la segunda tabla podemos aprovechar el elemento precedente $(a^{(t-1)\cdot s})$ y multiplicarlo por el primero (a^s).

Ejemplo 1.3. 1. Vamos a calcular el logaritmo en base 13 de 2 módulo 19, es decir, vamos a buscar b tal que $13^b = 2$ en $\mathbb{Z}_{19}$.

a) En primer lugar lo vamos a hacer probando. Empezamos a calcular las distintas potencias de 13.

- $13^2 = 13\cdot 13 = 17$.
- $13^3 = 13\cdot 17 = 12$.
- $13^4 = 13\cdot 12 = 4$.
- $13^5 = 13\cdot 4 = 14$.
- $13^6 = 13\cdot 14 = 11$.
- $13^7 = 13\cdot 11 = 10$.
- $13^8 = 13\cdot 10 = 16$.
- $13^9 = 13\cdot 16 = 18$.
- $13^{10} = 13\cdot 18 = 6$.
- $13^{11} = 13\cdot 6 = 2$.

Por tanto, el logaritmo vale 11.

b) Ahora vamos a utilizar el algoritmo de Shanks. Puesto que $\sqrt{19}$ está comprendido entre 4 y 5, tomamos $s = 5$.
Calculamos la primera tabla:

$$c = 2;\quad a\cdot c = 13\cdot 2 = 7;\quad a^2\cdot c = 13\cdot 7 = 15;\quad a^3\cdot c = 13\cdot 15 = 5;\quad a^4\cdot c = 5\cdot 13 = 8$$

Luego la primera tabla sería:

$$S = \begin{array}{|c|c|c|c|c|} \hline 2 & 7 & 15 & 5 & 8 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$$

Calculamos ahora la segunda:

$$a^5 = 14; \quad a^{10} = 14{\cdot}14 = 6; \quad a^{15} = 14{\cdot}6 = 8; \quad a^{20} = 14{\cdot}8 = 17; \quad a^{25} = 14{\cdot}17 = 10$$

La segunda tabla es entonces:

$$T = \begin{array}{|c|c|c|c|c|} \hline 14 & 6 & 8 & 17 & 10 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

Y encontramos una coincidencia entre la primera y segunda tabla en el número 8. Observando donde está, podemos ver que el logaritmo buscado es $3 \cdot 5 - 4 = 11$.

2. Vamos ahora a calcular el logaritmo en base 14 de 17, módulo 19. Procedemos como antes, y calculamos las dos tablas:

$$S = \begin{array}{|c|c|c|c|c|} \hline 17 & 10 & 7 & 3 & 4 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array} \qquad T = \begin{array}{|c|c|c|c|c|} \hline 10 & 5 & 12 & 6 & 3 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

Vemos que hay dos coincidencias 10 y 3. Por tanto, tenemos dos soluciones que son $b = 1 \cdot 5 - 1 = 4$ y $b' = 5 \cdot 5 - 3 = 22$.

El motivo de que nos hayan aparecido dos soluciones se debe a que el algoritmo nos da todos los posibles exponentes comprendidos entre 1 y 25 que son solución. Como $14^{18} = 1$, entonces si a cualquier solución le sumamos 18 obtenemos otra solución.

En este caso, al sumarle a 4 (una de las soluciones) 18 nos da un número comprendido entre 1 y 25, por tanto el algoritmo nos devuelve también esa solución.

3. Calculemos ahora $\log_9(15)(\text{mód } 19)$. Construimos las tablas:

$$S = \begin{array}{|c|c|c|c|c|} \hline 15 & 2 & 18 & 10 & 14 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array} \qquad T = \begin{array}{|c|c|c|c|c|} \hline 16 & 9 & 11 & 5 & 4 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

y vemos que no hay ninguna coincidencia, luego no existe el logaritmo.

4. Si hiciéramos $\log_9(11)(\text{mód } 19)$ nos saldría:

$$S = \begin{array}{|c|c|c|c|c|} \hline 11 & 4 & 17 & 1 & 9 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array} \qquad T = \begin{array}{|c|c|c|c|c|} \hline 16 & 9 & 11 & 5 & 4 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

y encontramos tres coincidencias que se corresponden con las soluciones 6, 15 y 24. El motivo es que $9^9 = 1$, y por tanto, a partir de una solución, sumándole 9 obtenemos otra.

Una forma de acortar el tiempo de ejecución del algoritmo es, en lugar de construir las dos tablas, construir sólo la primera. Una vez hecho esto, conforme vamos hallando los elementos de la segunda buscamos coincidencias con la primera, y en cuanto encontremos alguna, devolvemos la solución. De esta forma, además los requerimientos de memoria son menores, pues no es necesario almacenar los valores de la segunda tabla. Cada vez que calculemos uno, podemos borrar el anterior (salvo el primero). De esta forma, salvo en algunos casos extraños, vamos a conseguir la menor solución.

Ejemplo 1.4. Vamos a aumentar ahora el número primo, y vamos a calcular un logaritmo. Por ejemplo, calculemos $\log_{149}(184)(\text{mód } 211)$
En este caso tomamos como s el valor 15 ($14^2 < 211 < 15^2$).
La primera tabla sería:

$S =$

184	197	24	200	49	127	144	145	83	129	20	26	76	141	120
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Y ahora vamos calculando los elementos de la segunda tabla. Al hallar cada uno buscamos alguna coincidencia con alguno de la primera.
$149^{15} = 88$, que no coincide con ninguno de la primera. Seguimos:

$$149^{30} = 148;\ 149^{45} = 153;\ 149^{60} = 171;\ 149^{75} = 67;\ 149^{90} = 199;\ 149^{105} = 210;$$
$$149^{120} = 123;\ 149^{135} = 63;\ 149^{150} = 58;\ 149^{165} = 40;\ 149^{180} = 144.$$

Y aquí terminamos, pues el 144 lo tenemos en la primera tabla. La solución es entonces $180 - 6 = 174$.
Podemos comprobar como $149^{174} = 184$.

Método ρ de Pollard

Al igual que antes, dados a, b, p, con p un número primo, se trata de encontrar x tal que $a^x \equiv b(\text{mód } p)$. La idea es construir una sucesión $\{r_i\}_{i\in\mathbb{N}}$, y encontrar elementos que sean iguales dentro de la sucesión.
Para esto, dividimos $\mathbb{Z}_p^*$ en tres subconjuntos, aproximadamente del mismo tamaño, S_1, S_2 y S_3.
Elegimos un valor inicial $r_0 = (ab)^{x_0}$
La sucesión r_i se define:

$$r_{i+1} = \begin{cases} b \cdot r_i & \text{si } r_i \in S_1 \\ r_i^2 & \text{si } r_i \in S_2 \\ a \cdot r_i & \text{si } r_i \in S_3 \end{cases}$$

Notemos que r_i se puede escribir de la forma $r_i = a^{x_i} \cdot b^{y_i}$, donde $x_0 = y_0$ y

$$x_{i+1} = \begin{cases} x_i & \text{si } r_i \in S_1 \\ 2x_i(\text{mód } p-1) & \text{si } r_i \in S_2 \\ x_i + 1(\text{mód } p-1) & \text{si } r_i \in S_3 \end{cases} \qquad y_{i+1} = \begin{cases} y_i + 1(\text{mód } p-1) & \text{si } r_i \in S_1 \\ 2y_i(\text{mód } p-1) & \text{si } r_i \in S_2 \\ y_i & \text{si } r_i \in S_3 \end{cases}$$

Se busca un valor de i tal que $r_i = r_{2i}$
Una vez encontrado se tiene que $a^{x_i} \cdot b^{y_i} = a^{x_{2i}} \cdot b^{y_{2i}}$.
Si

$$m = y_i - y_{2i} (\text{mód } p-1) \qquad n = x_{2i} - x_i (\text{mód } p-1)$$

entonces tenemos que

$$a^n = b^m$$

Calculamos $d = mcd(m, p-1)$ y buscamos u, v tales que $d = m \cdot u + (p-1) \cdot v$. En tal caso, se tendría que $b^d = a^{n \cdot u}$

. 2

Tests de primalidad

En los criptosistemas de llave pública que vamos a estudiar los números primos son un elemento necesario. Por eso uno de los algoritmos que debemos conocer es aquél que nos permita saber si un número entero dado es primo o compuesto.
Podemos clasificar los test en dos grandes grupos:

1. Tests deterministas: Dado un número entero un test determinista nos dice con absoluta certeza si un número dado es primo o no.

2. Tests probabilísticos: Son tests pasados por todos los números primos, y la probabilidad de que un número compuesto pase el tests es tan pequeña como queramos.

Estos últimos son los más ampliamente utilizados en criptografía por la velocidad que tienen en comparación con los deterministas.
Vamos a describir tres de ellos: el test de Fermat, el test de Solovay-Strassen y el test de Miller-Rabin.
Los tres se basan en el teorema pequeño de Fermat.

. . . . . 2.1

Test de Fermat

En este caso se utiliza únicamente el teorema de Fermat para comprobar si el número es o no primo.
Supongamos que tenemos un número impar n y queremos ver si es o no primo. Entonces elegimos un número a comprendido entre 2 y $n-2$ y calculamos a^{n-1} en $\mathbb{Z}_n$.
Si el resultado no es 1, entonces podemos asegurar que el número no es primo.
Si el resultado es 1, entonces el número n podría ser primo. En tal caso, diremos que n es un *pseudoprimo para la base* a.
En el caso de que nos haya dado 1, para estar más seguros de que el número es primo, podemos repetir el test utilizando otro número b como base, y así tantas veces como queramos.
En general, se tiene que:

Sea n un número impar. Supongamos que existe a, tal que $mcd(a,n) = 1$ y $a^{n-1} \not\equiv 1(\text{mód } n)$. Entonces el número de bases para las que n es pseudoprimo es como mucho $\frac{\varphi(n)}{2}$.

Obviamente, si $mcd(a,n) \neq 1$ entonces n no puede ser pseudoprimo para la base a. De serlo, tendríamos que $a^{n-2} \cdot a = 1$ en $\mathbb{Z}_n$, lo que nos dice que a^{n-2} es un inverso de a módulo n, y sabemos que eso no es posible.

Ejemplo 2.1.

1. Consideramos el número $n = 15$. Este número, como sabemos, no es primo. Las bases para las que es pseudoprimo son:

   $$\{1, 4, 11, 14\}$$

   Del resto ($\{2, 3, 5, 6, 7, 8, 9, 10, 12, 13\}$) son primos relativos con 15 los siguientes:

   $$\{2, 7, 8, 13\}$$

   Es decir, de las 8 unidades que tiene $\mathbb{Z}_{15}$, para la mitad 15 es pseudoprimo, y para la otra mitad no lo es.

2. Sea ahora $n = 21$. En este caso, las bases para las que es pseudoprimo son:

   $$\{1, 8, 13, 20\}$$

   Como $\varphi(21) = 12$, hay 4 unidades de $\mathbb{Z}_{21}$ para las que es pseudoprimo y 8 para las que no.

3. Otro caso en el que el número de bases para las que es pseudoprimo es máximo es $n = 91 = 7 \cdot 13$. En este caso, $\varphi(n) = 72$, y hay 36 bases para las que es pseudoprimo:

   1,3,4,9,10,12,16,17,22,23,25,27,29,30,36,38,40,43,
   48,51,53,55,61,62,64,66,68,69,74,75,79,81,82,87,88,90

Obviamente, en el caso de que un número no sea primo lo mejor es que el número de bases para las que es pseudoprimo sea lo menor posible. Así será más improbable que demos con una base para la que sí lo es.
En el caso de que haya alguna unidad en $\mathbb{Z}_n$ para la que n no sea pseudoprimo, tenemos una cota superior para las bases en que sí lo es: $\frac{\varphi(n)}{2}$.
Pero puede darse el caso de que el número sea compuesto y no haya ninguna unidad para la que no sea pseudoprimo. Estos son los denominados *números de Carmichael*.
Dado un número n, se dice que n es un número de Carmmichael si n es compuesto y para cualquier a tal que $mcd(a,n) = 1$ se verifica que $a^{n-1} \equiv 1(\text{mód } n)$.

El número de Carmichael más pequeño es el 561. Este número se descompone como $561 = 3 \cdot 11 \cdot 17$.
El conjunto de unidades de $\mathbb{Z}_{561}$ es 320, lo que significa que hay 320 bases para las que es pseudoprimo y 240 para las que no. Eligiendo un número al azar entre 1 y 560, y pasando el test de Fermat para ese número tenemos probabilidad $\frac{4}{7} = 0'5714$ de que el test nos de un resultado erróneo. Esta probabilidad la podemos reducir un poco si eliminamos de las posibles bases al 1 y al $-1 = 560$ (que sabemos que siempre van a pasar el test). En ese caso, la probabilidad de que el test nos dé un resultado erróneo es $0'5699$.

. . . . . 2.2

Test de Solovay-Strassen

La idea de este test es muy parecida a la del test anterior. En este caso se usa el resultado de que si n es primo, entonces para cualquier a, $1 \leq a \leq n-1$ se verifica que

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) (\text{mód } n)$$

En el caso de que se verifique la identidad anterior se dice que n es un *pseudoprimo de Euler* para la base a.
Es claro que si n es pseudoprimo de Euler para la base a, entonces n es pseudoprimo para la base a. El recíproco no es cierto. Por ejemplo:

$$4^{45} \equiv 64 (\text{mód } 91) \quad \text{mientras que } 4^{90} \equiv 1 (\text{mód } 91)$$

Es decir, 91 es pseudoprimo para la base 4, pero no es pseudoprimo de Euler.
De las 36 base para las que 91 era pseudoprimo, sólo para la mitad es pseudoprimo de Euler. Éstas son:

1,9,10,12,16,17,22,29,38,53,62,69,74,75,79,81,82,90

En este caso sí podemos asegurar que si n es compuesto, entonces n es pseudoprimo para, como mucho, $\frac{n-1}{2}$ elementos de $\mathbb{Z}_n$.

. . . . . 2.3

Test de Miller-Rabin

El test anterior tiene un inconveniente. Requiere el cálculo de potencias modulares, así como del símbolo de Jacobi, lo que lo hace más lento.
El siguiente test, además del teorema pequeño de Fermat hace uso de que si n es primo, entonces en $\mathbb{Z}_n$ un polinomio de grado 2 tiene a lo sumo dos raíces. En particular, el polinomio $x^2 - 1$ no puede tener más de dos raíces. Como 1 y -1 son siempre raíces de ese polinomio, entonces no puede haber ningún otro número que al elevarlo al cuadrado nos de 1.
Antes de dar el test, veamos un ejemplo.
Sea $n = 145$, número que queremos ver si es o no primo. Elegimos una base, por ejemplo $a = 86$, y calculamos su símbolo de Jacobi.

Nos sale $\left(\frac{86}{145}\right) = 1$. Por otra parte, $86^{72} \equiv 1(\text{mód } 145)$. Significa esto que 145 es pseudoprimo de Euler para la base 86, es decir, pasa el test de Solovay-Strassen. Sin embargo, se tiene que:

$$86^{36} \equiv 1(\text{mód } 145) \qquad 86^{18} \equiv 1(\text{mód } 145) \qquad 86^{9} \equiv 86(\text{mód } 145)$$

De las relaciones $86^9 \equiv 86(\text{mód } 145)$ y $86^{18} \equiv 1(\text{mód } 145)$ podemos ver que 86 es una raíz cuadrada de la unidad en $\mathbb{Z}_{145}$. Si 145 fuera primo, la unidad sólo tendría dos raíces cuadradas en $\mathbb{Z}_{145}$, 1 y $144 = -1$. Por tanto, podemos deducir que 145 no es primo.
Esta es la idea del test de Miller-Rabin. Si n es nuestro candidato a primo, desomponemos $n-1$ como $2^u \cdot s$, con s un número impar (en el caso que acabamos de ver, $144 = 2^4 \cdot 9$). Elegimos una base para probar (en este caso $a = 86$), y vamos calculando, módulo n, las siguientes potencias:

$$a^s \qquad a^{2\cdot s} \qquad a^{2^2\cdot s} \qquad \cdots \qquad a^{2^{u-1}\cdot s} \qquad a^{2^u\cdot s}$$

Pueden ocurrir varias posibilidades:

1. Que la primera potencia, a^s valga 1 ó -1 (o si queremos, $n-1$). En tal caso, el número n podría ser primo.

2. Que ninguna de las potencias sea 1. En tal caso, el número n no es primo (en este caso no superaría el test de Fermat).

3. Que la primera potencia que dé 1 no venga precedida por -1. En tal caso n no es primo, pues tenemos una raíz cuadrada de 1 que no es ni 1 ni -1.

4. Que haya una potencia que de -1 (en cuyo caso la siguiente es 1). En tal caso, el número podría ser primo.

Notemos que, quitando la comprobación inicial ($a^s = \pm 1$)
- Si encontramos que una de las potencias vale -1 se acaba el proceso, y el número podría ser primo.
- Si encontramos que alguna potencia es 1, el número no puede ser primo.
- No es necesario calcular la última potencia ($a^{2^u\cdot s}$), pues si al llegar a la anterior no nos ha salido -1, el número sería compuesto.
Con esto, podemos dar el siguiente algoritmo que recoge todo lo dicho aquí:

Algoritmo MILLER-RABIN(p)

Entrada: $p \in \mathbb{Z} : p \geq 5$; p impar.

Salida: p no es primo o p es probable primo.

1. Expresamos $p-1$ como $2^u \cdot s$, con s un número impar.

2. Elegimos a al azar tal que $2 \leq a \leq p-2$.

3. Hacemos $a = a^s$.

4. Si $a = 1$ ó $a = -1$

a) Devuelve **p es probable primo.**

5. Si no

a) Desde $i = 1$ hasta $u - 1$

i $a := a^2$

ii Si $a = -1$

Devuelve **p es probable primo.**

iii Si $a = 1$

α Devuelve **p no es primo.**

iv $i := i + 1$.

b) Devuelve **p no es primo.**

Se tiene que si un número pasa el test de Miller-Rabin para una base, entonces para esa base pasa el test de Solovay-Strassen (y por tanto, también el de Fermat). El recíproco sabemos que no es cierto.

Si un número n es compuesto, el número de bases para las que pasa el test de Miller-Rabin es, como mucho, $\frac{n-1}{4}$. Por tanto, la probabilidad de que el test falle para un número compuesto es menor o igual que $\frac{1}{4}$. Si repetimos el test para m bases distintas, y el número que probamos es compuesto, la probabilidad de que falle para todos es menor o igual que $\frac{1}{4^m}$.

Por tanto, si tomamos un número, y le pasamos el test m veces, y en todas las ocasiones nos dice que podría ser primo, la probabilidad de que lo sea es mayor o igual que $\frac{4^m-1}{4^m}$. Basta hacer m lo suficientemente grande para que este valor lo aproximemos a 1 tanto como queramos. Por ejemplo, $\frac{4^{10}-1}{4^{10}} = 0'999999046$ (la posibilidad de que nos equivoquemos pasándolo 10 veces es menor que 1 entre 1000000). Para $m = 20$, la probabilidad de fallo es menor que 1 entre un billón, y para $m = 50$ la probabilidad de error es menor que 10^{-50}.

Ejemplo 2.2.

1. Vamos a tomar $n = 569$. Dividimos 568 entre dos hasta que nos de impar, y obtenemos $568 = 2^3 \cdot 71$. Elegimos distintas bases:

   a) $a = 478$. Entonces $a^{71} = 1$, luego 569 podría ser primo (con probabilidad mayor o igual que $\frac{3}{4}$.

   b) $a = 132$. En este caso se tiene que $a^{71} = -1$, luego la probabilidad de que sea primo es ahora mayor o igual que $\frac{15}{16}$.

   c) $a = 502$. Ahora $a^{71} = 86$. Calculamos entonces $a^{142} = 86^2 = -1$, luego vuelve a pasar el test. La probabilidad ahora es mayor o igual que $\frac{63}{64} = 0'984375$.

   De hecho, el número 569 es primo.

2. Tomamos ahora $n = 561$, que sabemos que no es primo. Tenemos que $560 = 2^4 \cdot 35$. Elegimos distintas bases:

   a) $a = 101$. Se tiene que $101^{35} = -1$. Por tanto, pasaría el test.

b) $a = 103$. Ahora resulta que $103^{35} = 1$. Por tanto, también pasa el test.

c) $a = 59$. Los cálculos son: $59^{35} = 155$; $59^{70} = 155^2 = 463$; $59^{140} = 67$; $59^{280} = 1$. Al llegar a 1 sin pasar por -1 deducimos que 561 no es primo. Notemos que en este caso, puesto que $\left(\frac{59}{561}\right) = -1 \neq 59^{280}$, entonces tampoco pasaría el test de Solovay-Strassen. Sin embargo, la comprobación del test de Solovay-Strassen es más costosa, pues requiere calcular el símbolo de Jacobi.

d) $a = 526$. Procediendo igual que antes, $529^{35} = 331$; $529^{70} = 166$; $529^{140} = 67$; $529^{280} = 1$. Por tanto, no pasa el test. En este caso sí pasaría el test de Solovay-Strassen, ya que $\left(\frac{529}{561}\right) = 1$.

e) $a = 62$. $62^{35} = 362$; $62^{70} = 331$; $62^{140} = 166$; $62^{280} = 67$. Ya no haría falta calcular 62^{560}, pues si esa potencia da 1 (que es lo que da), entonces habríamos obtenido un 1 sin pasar por -1 (de hecho, con lo que hemos calculado no pasaría el test de Solovay-Strassen), y si da distinto de 1 entonces no pasaría tampoco el test de Fermat.

Escribimos a continuación las distintas bases para las que 561 pasa el test de Fermat:

1	2	4	5	7	8	10	13	14	16	19	20	23	25	26	28
29	31	32	35	37	38	40	41	43	46	47	49	50	52	53	56
58	59	61	62	64	65	67	70	71	73	74	76	79	80	82	83
86	89	91	92	94	95	97	98	100	101	103	104	106	107	109	112
113	115	116	118	122	124	125	127	128	130	131	133	134	137	139	140
142	145	146	148	149	151	152	155	157	158	160	161	163	164	166	167
169	172	173	175	178	179	181	182	184	185	188	190	191	193	194	196
197	199	200	202	203	205	206	208	211	212	214	215	217	218	223	224
226	227	229	230	232	233	235	236	239	241	244	245	247	248	250	251
254	256	257	259	260	262	263	265	266	268	269	271	274	277	278	280
281	283	284	287	290	292	293	295	296	298	299	301	302	304	305	307
310	311	313	314	316	317	320	322	325	326	328	329	331	332	334	335
337	338	343	344	346	347	349	350	353	355	356	358	359	361	362	364
365	367	368	370	371	373	376	377	379	380	382	383	386	388	389	392
394	395	397	398	400	401	403	404	406	409	410	412	413	415	416	419
421	422	424	427	428	430	431	433	434	436	437	439	443	445	446	448
449	452	454	455	457	458	460	461	463	464	466	467	469	470	472	475
478	479	481	482	485	487	488	490	491	494	496	497	499	500	502	503
505	508	509	511	512	514	515	518	520	521	523	524	526	529	530	532
533	535	536	538	541	542	545	547	548	551	553	554	556	557	559	560

que son 320, exactamente todas las unidades de $\mathbb{Z}_{561}$.
De esos 320, pasarían el test de Solovay-Strassen los 84 siguientes:

1	2	4	8	16	25	31	32	35	49	50	64	67	70
83	98	100	101	103	115	128	134	140	149	157	161	166	169
196	200	202	206	218	223	229	230	239	247	256	263	268	280
281	293	298	305	314	322	331	332	338	343	355	359	361	365
392	395	400	404	412	421	427	433	446	458	460	461	463	478
491	494	497	511	512	526	529	530	536	545	553	557	559	560

Y de estos 84, los que pasarían el test de Miller-Rabin son:

$$1 \quad 50 \quad 101 \quad 103 \quad 256 \quad 458 \quad 460 \quad 511$$

Vemos aquí como la probabilidad de error que se ha dado es una cota superior. En este caso, la probabilidad de que falle el test de Solovay-Strassen es $\frac{84}{560} = \frac{3}{30} = 0'15$, que es muy inferior al valor dado de $0'5$. Si además eliminamos de las posibles bases 1 y $560 = -1$ (pues esas siempre van a pasar el test), nos queda entonces $\frac{82}{558} = \frac{41}{279} = 0,14695$.
Para el test de Miller-Rabin, para este número la probabilidad de error es $\frac{8}{560} = \frac{1}{70} = 0'014$, que se reduciría a $\frac{6}{558} = \frac{1}{93} = 0'01075$ si eliminamos las bases 1 y 560.

3 El problema de la factorización

De todos es conocido que todo número natural mayor que 1, o es primo, o se puede expresar de forma única como producto de primos. Para obtener la expresión de un número como producto de primos, podemos ir divididiendo por los primos pequeños, y cuando encontremos que uno lo divide, continuar el proceso con el cociente.
Por ejemplo, si tomamos $n = 7260$, sabemos que es múltiplo de 2 pues acaba en cifra par (el cero), luego lo dividimos entre 2: $7260 = 2 \cdot 3630$
Este número también es múltiplo de 2, luego volvemos a dividirlo por 2, y nos queda que $3630 = 2 \cdot 1815$, luego $7260 = 2^2 \cdot 1815$.
Este número, al terminar en cifra impar no es dividible por 2. Probamos entonces por 3, y como la suma de sus cifras es 15, que es múltiplo de 3, el número también lo es. Haciendo la división obtenemos que $7260 = 2^2 \cdot 3 \cdot 605$.
El número 605 no es múltiplo de 3, pues la suma de sus cifras es 11 (que no es múltiplo de 3), pero sabemos que es múltiplo de 5 por acabar en 5. Tenemos entonces que $7260 = 2^2 \cdot 3 \cdot 5 \cdot 121$.
El 121 no es divisible por 5, ni por 7 ($121 = 7 \cdot 17 + 2$), pero sí es múltiplo de 11 (la suma de las cifras que ocupan posición par, menos la suma de las cifras que ocupan posición impar es 0 o múltiplo de 11), y de aquí podemos obtener que

$$7260 = 2^2 \cdot 3 \cdot 5 \cdot 11^2$$

y ya tenemos la expresión de 7260 como producto de números primos.
El proceso que hemos seguido podemos hacerlo fácilmente con lapiz y papel para números no muy grandes. Incluso, para números grandes, si los divisores primos son pequeños puede hacerse rápidamente. Por ejemplo, sea $n = 7009695$.

1. El número no es múltiplo de 2. Pero de 3 sí lo es, pues $7+9+6+9+5 = 36$ es múltiplo de 3. Dividimos entonces por 3 las veces que podamos.

$$n = 3^2 \cdot 778855$$

2. Este número es múltiplo de 5. Tenemos entonces $n = 3^2 \cdot 5 \cdot 155771$.

3. Probamos a dividir por 7, y nos queda $n = 3^2 \cdot 5 \cdot 7 \cdot 22253 = 3^2 \cdot 5 \cdot 7^2 \cdot 3179$, y paramos pues 3179 no es múltiplo de 7.

4. El número 3179 es múltiplo de 11, pues $(9 + 1) - (7 + 3) = 0$. Se tiene entonces $n = 3^2 \cdot 5 \cdot 7^2 \cdot 11 \cdot 289$.

5. 289 no es divisible por 13, pero sí lo es por 17. Al hacer esta división nos queda:

$$7009605 = 3^2 \cdot 5 \cdot 7^2 \cdot 11 \cdot 17^2$$

y ya hemos completado su factorización.
Sin embargo, si los divisores primos no son pequeños, el proceso se alarga bastante. Por ejemplo, sea $n = 7011461$ (que se encuentra muy próximo al que acabamos de factorizar). En este caso, la factorización de n como producto de primos es $n = 1931 \cdot 3631$.
En esta ocasión tendríamos que haber probado por todos los primos $2, 3, 5, \cdots$ hasta encontrar alguno que lo dividiera (el 1931). Este lo habríamos encontrado después de 293 intentos fallidos (el primo 1931 ocupa el ducentésimo nonagésimo cuarto lugar en la lista de los números primos).
No obstante, si dispusiéramos de una lista con los primeros números primos (los primeros 1000, los primeros 10000, el primer millón, etc.) estos cálculos serían fácilmente realizables con los medios actuales.
Pero cuando el tamaño del número n a factorizar, o mejor dicho, el tamaño de los primos que intervienen en su factorización aumenta, estos cálculos son demasiado lentos incluso para potentes ordenadores.
Vamos a ver a continuación algunos algoritmos para calcular la factorización de un número.

. . . . . 3.1

División por tentativa

Este es el algoritmo que acabamos de describir. Se trata de ir dividiendo el número a factorizar por los diferentes primos, hasta encontrar alguno que lo divida. Para terminar, puede ser útil el siguiente resultado:

> Sea n un número natural. Si n es compuesto, entonces n tiene un factor primo menor o igual que $\sqrt{n}$.

Por ejemplo, si $n = 7011461$ entonces $\sqrt{n} = 2647'9$.
Esto es claro, pues si todos los factores primos fueran mayores que $\sqrt{n}$, su producto sería mayor que n. El mayor primo anterior a dicho valor es 2647, que ocupa la posición número 383 en la lista de números primos. Por tanto, para factorizar n tendríamos que ver si n es múltiplo de 383 números. Si no lo es de ninguno de ellos, n sería primo. En caso contrario, ya tenemos una factorización de n. Sabemos que el número n no es primo, y que esto lo podemos saber después de 294 comprobaciones.

3.2 *Método de Fermat*

La idea de este método es expresar el número a factorizar como diferencia de dos cuadrados, es decir, $n = x^2 - y^2$. En tal caso, una factorización de n sería $n = (x+y)\cdot(x-y)$. Obviamente, los números x e y no pueden ser $\frac{n+1}{2}$ y $\frac{n-1}{2}$. La factorización que se obtendría entonces sería $n = n \cdot 1$.
El método que vamos a describir es útil en el caso de que n tenga dos divisores relativamente próximos, y próximos a $\frac{n}{2}$. Entonces procederíamos como sigue:
Calculamos la raíz cuadrada de n, y llamamos x al entero inmediatamente superior.
Mientras $x^2 - n$ no sea un cuadrado perfecto, incrementamos x en una unidad.
Cuando sea un cuadrado perfecto, le damos a y el valor igual a la raíz cuadrada de $x^2 - n$.
Con los valores de x e y conseguimos una factorización de n.

Ejemplo 3.1. Sea $n = 6352351$. Vamos a factorizarlo usando este método.
Se tiene que $\sqrt{n} = 2520'387$. Por tanto, le damos a x el valor 2521. Nos quedaría:

x	$x^2 - n$	¿es $x^2 - n$ cuadrado perfecto?
2521	3090	NO
2522	8133	NO
2523	13178	NO
2524	18225	SÍ

Le damos a y el valor $\sqrt{18225} = 135$, y nos queda:

$$6352351 = (2524 + 135) \cdot (2524 - 135) = 2659 \cdot 2389$$

Por ejemplo, si tuviéramos $n = 259824356131$ necesitaríamos 100 iteraciones para obtener una factorización de n.

3.3 *Algoritmo ρ de Pollard*

Este algoritmo fue propuesto por John M. Pollard en 1975 para encontrar un factor no trivial de un número n.
La idea es, dado un número n, construir una sucesión $x_0, x_1, \cdots, x_m, \cdots$ de elementos de $\mathbb{Z}_n$. Para eso, nos valemos de una función auxiliar $f : \mathbb{Z}_n \to \mathbb{Z}_n$, y de un valor inicial o semilla x_0, y definimos la sucesión recursivamente mediante la regla $x_n = f(x_{n-1})$.
Si d es un divisor de n distinto de 1 y de n, es probable que en la sucesión obtenida existan términos x_i, x_j tales que $x_i \equiv x_j$ mód d pero que $x_i \neq x_j$ (módulo n). En tal caso, tenemos que d es un divisor de $x_i - x_j$, y también un divisor de n, luego $mcd(x_i - x_j, n)$ es un divisor de n.
Por ejemplo, tomamos $n = 187$, y $f(x) = x^2 + 3$. A partir del valor $x_0 = 7$ construimos la sucesión:

$$\begin{array}{llllll} x_0 = 7 & x_1 = 52 & x_2 = 89 & x_3 = 70 & x_4 = 41 & x_5 = 1 \\ x_6 = 4 & x_7 = 19 & x_8 = 177 & x_9 = 103 & x_{10} = 140 & x_{11} = 155 \\ x_{12} = 92 & x_{13} = 52 & x_{14} = 89 & \cdots & & \end{array}$$

Y a partir de aquí se repiten.
Podemos comprobar como $x_4 - x_0 = 34$, que es múltiplo de 17, al igual que 187. Por tanto, $mcd(x_4 - x_0, n) = 17$, y ya tenemos un factor del número n.
También se tiene que $x_4 - x_1 = -11$, luego $mcd(x_4 - x_1, n) = 11$, y aquí ya tenemos otro factor.
Es decir, hemos encontrado términos de la sucesión, por ejemplo x_0 y x_4, que son distintos, pero que cumplen que $mcd(x_4 - x_0, n) \neq 1$. Y esto nos da un factor del número n. De la misma forma, podríamos haber tomado los términos x_1 y x_4.
Notemos que puesto que $x_4 - x_0$ es múltiplo de 17, también es múltiplo de 17 $x_5 - x_1$, y $x_6 - x_2$, etc. Igual ocurre con los términos x_1 y x_4. Al ser $x_4 - x_1$ múltiplo de 11, también es múltiplo de 11 $x_5 - x_2$, $x_6 - x_3$, etc. Sin embargo, $x_4 - x_0$ es múltiplo de 2, pero no lo es $x_5 - x_1$. ¿Podrías explicar porqué ocurre esto?
Aquí encontramos una primera aproximación al algoritmo ρ de Pollard.

> Comenzamos con una función $f : \mathbb{Z}_n \to \mathbb{Z}_n$, y un elemento aleatorio $x_0 \in \mathbb{Z}_n$.
>
> Calculamos $x_1 = f(x_0)$.
>
> Calculamos $mcd(x_1 - x_0, n)$. Si vale distinto de 1, ya tenemos un divisor de n. Si vale 1, continuamos.
>
> Calculamos $x_2 = f(x_1)$.
>
> Calculamos $mcd(x_2 - x_1, n)$ y $mcd(x_2 - x_0, n)$. Si alguno vale distinto de 1, terminamos. Si ambos valen 1, continuamos.
>
> Calculamos $x_3 = f(x_2)$.
>
> Calculamos $mcd(x_3 - x_2, n)$, $mcd(x_3 - x_1, n)$ y $mcd(x_3 - x_0, n)$. Y repetimos lo mismo que en los casos anteriores.

Aún cuando para el cálculo del máximo común divisor disponemos de un buen algoritmo, como es el algoritmo de Euclides, el método que acabamos de describir necesita llamar a este algoritmo muchas veces.
Para reducir esto, vamos a fijarnos en el ejemplo que hemos visto.
De acuerdo con este ejemplo, necesitaríamos calcular $mcd(x_1 - x_0, n)$; $mcd(x_2 - x_1, n)$; $mcd(x_2 - x_0, n)$; $mcd(x_3 - x_2, n)$; $mcd(x_3 - x_1, n)$; $mcd(x_3 - x_0, n)$; $mcd(x_4 - x_3, n)$; $mcd(x_4 - x_2, n)$. Y en todos los casos obtendríamos 1 como resultado. Al realizar el siguiente cálculo, es decir, $mcd(x_4 - x_1, n)$ obtendríamos como resultado 11, lo que nos daría un divisor del número n.
Sin embargo, la observación que hicimos justo antes de dar la primera aproximación al algoritmo, nos dice que podríamos haber encontrado el divisor 11 calculando $mcd(x_5 - x_2, n)$ o bien $mcd(x_6 - x_3, n)$. En este último es en el que nos vamos a fijar, y lo que vamos a ir

haciendo es calcular $mcd(x_{2m} - x_m, n)$. Entonces vamos a ir calculando dos sucesiones. La sucesión x_m, tal y como hemos dicho antes, y la sucesión $y_m = x_{2m}$.
El algoritmo podría quedar entonces

> Comenzamos con una función $f : \mathbb{Z}_n \to \mathbb{Z}_n$, y un elemento aleatorio $x_0 \in \mathbb{Z}_n$.
>
> Calculamos $x_1 = f(x_0)$, e $y_1 = f(x_1)$.
>
> Calculamos $mcd(y_1 - x_1, n)$. Si vale distinto de 1, ya tenemos un divisor de n. Si vale 1, continuamos.
>
> Calculamos $x_2 = f(x_1)$ e $y_2 = f(f(y_1))$.
>
> Calculamos $mcd(y_2 - x_2, n)$. Si vale distinto de 1, terminamos. Si vale 1, continuamos.
>
> Calculamos $x_3 = f(x_2)$ e $y_3 = f(f(y_2))$.
>
> Calculamos $mcd(y_3 - x_3, n)$. Al igual que antes, si vale distinto de 1, terminamos. Si vale 1, continuamos.

De esta forma, además el gasto de memoria es mucho menor, pues no es necesario almacenar los términos de la sucesión, algo que sí requeríamos en la primera versión que dimos del algoritmo.
Por otra parte, como función f se suele emplear la función dada por $f(x) = x^2 + 1$.
Obviamente, al algoritmo hay que darle una condición de parada. Esta puede darse fijando previamente el número máximo de iteraciones. Vamos a denotar a este número como I. Si queremos encontrar una factorización del número n, el algoritmo quedaría entonces:

- Elegimos $a \in \mathbb{N}$ aleatorio, de forma que $1 \leq a \leq n - 1$.
- $x := a^2 + 1$ e $y := x^2 + 1$ (ambos cálculos se realizan módulo n).
- Desde $i = 1$ hasta I
  - $d := mcd(y - x, n)$
  - Si $d = n$ Devuelve n es probable primo.
  - Si $d = 1$
    - $i = i + 1$.
    - $x := x^2 + 1$ (módulo n).
    - $y := (y^2 + 1)^2 + 1$ (módulo n).
    - Volvemos al principio del bucle.
  - Devuelve d *es un divisor de* n.
- Devuelve n *es probable primo.*

Vamos a ver algunos ejemplo.
En primer lugar, probamos con $n = 187$.

i	a	x	y	$y-x$	$mcd(y-x,n)$
	7				
1		52	89	37	1
2		89	41	−48	1
3		70	4	−66	11

Y vemos como 11 es un divisor.
Tomamos ahora $n = 40259$. Si le pasamos, por ejemplo, el test de Miller-Rabbin, vemos que no es primo. Entonces vamos a factorizarlo siguiendo el algoritmo ρ de Pollard.

i	a	x	y	$y-x$	$mcd(y-x,n)$
	5975				
1		31152	3910	−27242	1
2		3910	36966	33056	1
3		29940	30855	915	1
4		36966	19531	−17435	317

Y ya tenemos una factorización: $40259 = 317 \cdot 127$.

. . . . . 3.4

Algoritmo de Strassen.

La idea de este algoritmo es similar a la del algoritmo del *Paso enano, paso gigante* que vimos para calcular el logaritmo discreto.
Supongamos que tenemos un número n, y queremos encontrar un factor suyo menor que una cota b (podemos tomar $b = \lfloor\sqrt{n}\rfloor$, es decir, la parte entera de $\sqrt{b}$). Si el número n no es primo, tendrá un divisor menor que esa cota).
Dividimos el intervalo en el que queremos encontrar el factor en bloques de un tamaño c. Este número c podemos tomarlo como $\lceil\sqrt{b}\rceil$ (es decir, el primer entero por encima de $\sqrt{b}$). Es decir, el intervalo $[1, b]$, lo tenemos dividido en subintervalos $[1, c]$, $[c+1, 2c]$, $\cdots [c^2 - c + 1, c^2]$.
Ahora, miramos si n tiene un divisor en cada uno de esos subintervalos.
Para esto, consideramos en $\mathbb{Z}_n$ el polinomio $f(x) = \prod_{k=1}^{c}(x+k) = (x+1)(x+2)\cdots(x+c)$.
Evaluamos el polinomio en $x = 0$, $x = c$, $x = 2c$, $\cdots x = (c-1)c$. Es decir, calculamos $g_i = f(i \cdot c)$ para $i = 0, 1, \cdots, c-1$.
Calculamos $mcd(g_i, n)$. Si para algún i se tiene que $mcd(g_i, n) \neq 1$, ya tenemos un divisor de n en el intervalo $i \cdot c + 1, \cdots (i+1) \cdot c$.
Por ejemplo. Tomamos $n = 551$. Si n no es primo, debe tener un divisor menor que $\sqrt{551} = 23'47$. Por tanto, buscamos un divisor de n menor que 23. En este caso, $c = \lceil\sqrt{23}\rceil = \lceil 4'79\rceil = 5$.
Sea $f(x) = (x+1)(x+2)(x+3)(x+4)(x+5) = x^5 + 15x^4 + 85x^3 + 225x^2 + 274x + 120$.
Evaluamos: $f(0) = 120$; y calculaomos $mcd(120, 551) = 1$. Por tanto, continuamos.

Evaluamos: $f(5) = 486$; y calculamos $mcd(486, 551) = 1$. Continuamos.
Evaluamos: $f(10) = 6$; y calculamos $mcd(6, 551) = 1$. Seguimos.
Evaluamos: $f(15) = 304$; y calculamos $mcd(304, 551) = 19$.
Con esto, ya hemos encontrado un divisor de n. El 19. Puesto que $f(15) = 16 \cdot 17 \cdot 18 \cdot 19 \cdot 20$, el divisor se tiene que encontrar en ese intervalo.
Si en lugar de $n = 551$ hubiéramos tomado $n = 557$ (que es primo), habríamos obtenido los siguientes resultados:
$b = 23$, $c = 5$, $f(x) = x^5 + 15x^4 + 85x^3 + 225x^2 + 274x + 120$.
$f(0) = 120$, $mcd(120, 557) = 1$.
$f(5) = 162$, $mcd(162, 557) = 1$.
$f(10) = 532$, $mcd(532, 557) = 1$.
$f(15) = 100$, $mcd(100, 557) = 1$.
$f(20) = 178$, $mcd(178, 557) = 1$.
Lo que nos dice que 557 no tienen ningún divisor menor que 25. Por tanto, 557 es un número primo.
Ordenando estas ideas, podemos describir el algoritmo, que tiene dos entradas, n y b, y nos devuelve, bien que n no tiene divisores menores que b (salvo 1), bien un número k, divisor de n. En el caso de que tomemos $b = \lfloor \sqrt{n} \rfloor$, el algoritmo nos dice si n es primo, o no, y en caso de que no lo sea, nos da un divisor suyo.

- $c := \lceil \sqrt{b-1} \rceil$.
- $p(x) := 1$.
- Desde $k = 1$ hasta c
  - $p(x) := (x + k) \cdot p(x)$ (módulo n).
  - $k = k + 1$.
- Desde $i = 0$ hasta $c - 1$.
  - $g := h(c \cdot i)$ (módulo n).
  - $d := mcd(g, n)$.
  - Si $d \neq 1$, hacemos $i = c - 1$.
- Si $d = 1$ devuelve "n *no tiene divisores menores que* b. *En caso contrario, devuelve* d.

Vamos a tomar $n = 1357$, y vamos a ver si es o no primo, y en caso negativo, vamos a encontar un divisor suyo. Para eso, hacemos $b = \lfloor \sqrt{1357} \rfloor = 36$.
Se tiene que $c = \lceil \sqrt{b-1} \rceil = \lceil \sqrt{35} \rceil = 6$. Calculamos primero el polinomio $p(x)$.

k	p(x)
1	$x+1$
2	x^2+3x+2
3	$x^3+6x^2+11x+6$
4	$x^4+10x^3+35x^2+50x+24$
5	$x^5+15x^4+85x^3+225x^2+274x+120$
6	$x^6+21x^5+175x^4+735x^3+267x^2+407x+720$

Y una vez calculado, entramos en el cuerpo del algoritmo.

i	g	d
0	720	1
1	350	1
2	987	1
3	322	23

Por tanto, 1357 no es primo, y 23 es un divisor suyo. De hecho, se tiene que $1357 = 23 \cdot 59$.
Podría parecer que este algoritmo nos devuelve el primer divisor del número n (siempre y cuando sea menor que b). Sin embargo, esto no es así. El algoritmo, en el caso de que haya un divisor de n menor que b, nos encuentra un divisor de n que es múltiplo del menor divisor de n. Por ejemplo, si tomamos $n = 690$, y $b = \lfloor\sqrt{690}\rfloor = 26$, al aplicar el algoritmo nos devuelve 30.
Podemos comprobar como 30 es un divisor de 690, y es múltiplo del menor divisor de 690, que es 2.

Existen otros muchos métodos de factorización, pero todos son lentos cuando se trata de factorizar números grandes, salvo que los factores tengan alguna propiedad (por ejemplo, en los que hemos visto que los factores sean pequeños, o que estén próximos entre sí). Sin embargo hemos visto que existen formas para determinar si un número es o no primo, con un margen de error ínfimo.

. . . . . 3.5

Raíz cuadrada y factorización.

Hemos visto que si p es un número primo podemos determinar si un elemento $a \in \mathbb{Z}_p$ tiene o no raíz cuadrada, y en caso afirmativo calcularla. El problema varía cuando el módulo en el que trabajamos no es primo. Vamos a centrarnos aquí en el caso de que el módulo sea producto de dos números primos impares p y q.
Sean p y q dos números primos, y $n = p \cdot q$.
Notemos que si un número a, tal que $mcd(a, n) = 1$ tiene raíz cuadrada módulo n, entonces tiene exactamente 4 raíces cuadradas. Si b es una raíz cuadrada de a módulo n, entonces $b^2 \equiv a(\text{mód } p)$ y $b^2 \equiv a(\text{mód } q)$. En ese caso, las soluciones a los cuatro siguientes sistemas de congruencias es una raíz cuadrada de a módulo n.

$$\begin{array}{llll} x \equiv b(\text{mód } p) & x \equiv b(\text{mód } p) & x \equiv p-b(\text{mód } p) & x \equiv p-b(\text{mód } p) \\ x \equiv b(\text{mód } q) & x \equiv q-b(\text{mód } q) & x \equiv b(\text{mód } q) & x \equiv q-b(\text{mód } q) \end{array}$$

La solución a la primera es b, la solución a la cuarta es $n-b$, y las soluciones de la segunda y tercera suman también n.
Es decir, la cuarta parte de los elementos de $\mathbb{Z}_p^*$ tienen raíz cuadrada, y las tres cuartas partes restantes no tienen.
Vemos además que las raíces cuadradas van por parejas. Por una parte tenemos b y $n-b$, y por otra parte tenemos otra raíz c y $n-c$. A dos raíces cuadradas (distintas) cuya suma sea distinta de n las llamaremos raíces gemelas.
Hemos visto aquí que si conocemos la factorización del número n, entonces podemos calcular las cuatro raíces cuadradas de un elemento (si es que las tiene). Basta para ello calcular las raíces cuadradas módulo p y módulo q y resolver los correspondientes sistemas de congruencias.
Al revés también es cierto. Si conocemos las cuatro raíces cuadradas de un número a, entonces podemos calcular la factorización de n.
Supongamos que b y c son dos raíces cuadradas gemelas de a (podemos elegir ambas que sean menores que $\frac{n}{2}$). En ese caso, $b^2 \equiv c^2(\text{mód } n)$, luego b^2-c^2 es múltiplo de n. Por tanto, $(b+c)\cdot(b-c)$ es múltiplo de n. Como n no es múltiplo de $b+c$ ni de $b-c$, entonces los factores de n tienen que estar repartidos, uno en $b+c$ y el otro en $b-c$. En ese caso, basta calcular $mcd(b+c,n)$.
Por ejemplo, podemos ver cómo 362902, 512256, 807977 y 957311 son raíces cuadradas de 659155 módulo 1320233. Tomamos $b = 362902$ y $c = 512256$, que son gemelas. Se tiene que:

$$mcd(b+c, 1320233) = mcd(875158, 1320233) = 937$$

Por tanto, 937 es un divisor de 1320233. Si dividimos ahora 1320233 entre 937 obtenemos que $1320233 = 937 \cdot 1409$.
Es decir, dado un número n producto de dos primos, si conocemos su factorización podemos hallar las raíces cuadradas módulo n, y si conocemos las raíces cuadradas en $\mathbb{Z}_n^*$ de un elemento, entonces podemos conocer sus divisores primos.
Si además $p \equiv 3(\text{mód } 4)$ y $q \equiv 3(\text{mód } 4)$, y b y c son dos raíces cuadradas de un mismo elemento, entonces:

- Si b y c son gemelas, entonces $\left(\frac{b}{n}\right) = -\left(\frac{c}{n}\right)$.
- Si b y c no son gemelas, entonces $\left(\frac{b}{n}\right) = \left(\frac{c}{n}\right)$.

Esto último es consecuencia de que $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$.

Parte V

Aplicaciones criptográficas

4

Introducción.

Hasta el momento hemos descrito algunos de los criptosistemas más usuales. Tanto de cifrado en flujo como de cifrado en bloque. Y dentro de estos últimos hemos visto simétricos y asimétricos. Hay muchos que se han quedado fuera, pero los vistos hasta ahora nos parecen suficientes para cubrir el objetivo de estas notas.

Con los distintos criptosistemas que hemos detallado sabemos cómo hacer que una comunicación sea confidecial, es decir, que sólo conozcan su contenido las personas autorizadas para ello. Y si alguien fuera capaz de interceptar el mensaje no sería capaz de entender lo que se está transmitiendo.

Es decir, hemos tratado el tema de la *confidencialidad*. Pero hay otras muchas situaciones y problemas que se pueden plantear en las comunicaciones digitales, y que con ayuda de las técnicas criptográficas estudiadas, y algunas que veremos en su momento se pueden resolver. En lo que sigue vamos a estudiar algunos de estos problemas y algunas de las soluciones que se han propuesto. No pretendemos hacer un estudio exhaustivo, sino dar algunas pinceladas que permitan al lector encontrarse en disposición de abordar en un futuro problemas relacionados con la criptografía.

CRIPTOGRAFÍA XII

Autenticación y firmas digitales.

..... 1

Introducción.

El rápido avance tecnológico en que estamos inmersos está provocando un cambio en muchos hábitos de vida. Muchas acciones que para hacerlas teníamos que presentarnos físicamente en algún lugar, o que realizábamos telefónicamente (declaración de la renta, compra, transacciones bancarias, etc.), hoy en día podemos hacerlas sentados delante de un ordenador. Nuestros datos van a viajar a través de la red de redes, y si no tomamos las precauciones necesarias, estos datos podrían ser leídos, borrados o modificados con el consiguiente perjuicio para nuestros intereses.

En este capítulo vamos a centrarnos en la firma de documentos. Notemos que cuando firmamos un documento éste no tiene porqué ser confidencial, sino que se buscan otros atributos.

Vamos a ver cuales son las características de una firma tradicional, y los usos que tiene, y eso nos dirá cuál es su análogo digital.

Cuando firmamos un documento, damos conformidad a lo que allí está escrito. A lo que está escrito en su totalidad. Una modificación sería detectable, pues deja huellas físicas, y el firmante podría negar haberlo firmado. Por ejemplo, cuando yo entrego el acta de calificaciones de la asignatura de Criptografía, tengo que firmar ese acta, y con eso las calificaciones quedan como están en el documento. Si tuviera que hacer una modificación tendría que volver a firmar el documento indicando la modificación realizada. De esta forma, el documento válido es el documento entero, con su modificación. Esto es lo que se conoce como *integridad*. El documento firmado no ha sido alterado. La firma de un documento garantiza su integridad.

Otra característica de un documento firmado es que el autor de la firma no puede negar que la ha hecho. La firma es algo inherente a la persona, y de negar alguien la autoría de la firma hay procedimientos de análisis para comprobar la veracidad o falsedad de tal afirmación. Esto es lo que se conoce como *no repudio*.

Obviamente, por lo dicho anteriormente la firma es infalsificable. Al tener la firma unas características propias de la persona que firma, nadie podría suplantar esa firma. Y precisamente por esto, la firma es verificable. Se puede comprobar si una firma es de quien dice ser o no.

Cuando firmamos un documento lo que hacemos es añadirle la firma, de forma que ambos, documento y añadido, forman un todo. Este añadido es el mismo (o similar) para todos los documentos que firmamos.

En seguida podemos ver que esto no va a ser posible para documentos digitales. En este caso, el añadido deber ser un conjunto de bits, y si estos son los mismos para todos los documentos firmados por un usuario, cualquiera podría firmar un documento por él.
Por tanto, la firma digital debe ser un añadido al documento que sólo pueda ser realizado por el firmante, pero que además debe variar con el mensaje, es decir, debe depender del documento.
La ley 59/2003, referente a la firma electrónica afirma en su artículo 3.1 que *La firma electrónica es el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante*, y en su artículo 3.4 que *la firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica el mismo valor que la firma manuscrita en relación con los consignados en papel.*
Algunas características que debe satisfacer la firma digital, y que ya hemos avanzado, son:

- *Únicas.* La firma sólo puede ser generada por el firmante. Si alguien quisiera suplantar una firma, tendría que resolver problemas de muy elevada complejidad.
- *Verificables.* Una firma debe ser fácilmente verificable, tanto por el receptor como por jueces.
- *No repudiables.* El firmante no puede negar haber realizado la firma.
- *Viables.* La firma debe ser fácil de realizar.

Podemos definir un sistema de firma como una quíntupla $(\mathcal{M}, \mathcal{F}, \mathcal{K}, \mathrm{fir}, ver)$ donde:

1. $\mathcal{M}$ es el conjunto de posibles mensajes;
2. $\mathcal{F}$ es el conjunto de posibles firmas;
3. $\mathcal{K}$ es el conjunto de posibles claves;
4. fir es una función $\mathrm{fir} : \mathcal{M} \longrightarrow \mathcal{F}$, llamada función firma;
5. ver es una función $ver : \mathcal{M} \times \mathcal{F} \longrightarrow \mathbf{Z}_2$ definida por:

$$ver(x, y) = \begin{cases} 1 & \text{si } y = \mathrm{fir}(x) \\ 0 & \text{si } y \neq \mathrm{fir}(x) \end{cases}$$

Seguidamente entraremos a describir algunos sistemas de firma digital:

2 MAC

Este sistema hace uso de la criptografía simétrica. En concreto al sistema de cifrado DES, aunque podría ser válido para cualquier otro sistema de cifrado por bloques simétrico.
Las siglas MAC hacen referencia a *Message Authentication Code*. Dado un mensaje, se le añade el MAC que garantiza que el mensaje no ha sido modificado.
Para calcular el MAC se cifra el mensaje con DES, utilizando el modo CBC (o CFB) con una clave que comparten el emisor y el receptor. Como vector de inicialización se toma el $(0, 0, \cdots, 0)$. Del último bloque se extraen los últimos 16 ó 32 bits (o el bloque completo), y eso constituye el MAC.
El emisor envía entonces el mensaje junto con el MAC calculado.
El receptor, para comprobar la integridad del mensaje procede de igual forma. Toma el mensaje, y calcula el MAC y lo compara con el que ha recibido. Caso de coincidir se considera el mensaje auténtico.
Notemos que si en lugar de usar el modo CBC hubiéramos empleado el ECB entonces aun atacante podría modificar el mensaje en cualquiera de los bloques salvo el último, pues esta modificación no afectaría al MAC. Al emplear el modo CBC una modificación en un bloque afecta a todos los que le siguen, en concreto al último, y al no disponer de la clave no podría calcular el MAC.

3 Funciones Resumen

3.1 *Generalidades*

En el apartado anterior ha aparecido una idea muy usada en los procesos de firma digital. A un mensaje se le añade un bloque que es del mismo tamaño, independientemente del tamaño mensaje. Si esto no fuera así, el mensaje cifrado podría tener gran tamaño. Además, para la generación de firmas suele usarse la criptografía asimétrica que en general es bastante lenta, luego no es práctico, ni para generarlas ni para verificarlas, disponer de gran cantidad de información.
En el caso anterior el bloque añadido depende tanto del mensaje como de una clave. En las funciones resumen que vamos a estudiar en esta sección no es necesario el uso de ninguna clave.
Una función resumen (o función hash, del inglés "pìcar y mezclar") es una función de una sola dirección resistente a colisiones. Estas funciones transforman cada posible mensaje en un resumen de un tamaño determinado (por ejemplo, 128 ó 160 bits).
Dado que el cardinal del conjunto de posibles mensajes es mayor que el cardinal de resúmenes, tiene que haber diferentes mensajes con el mismo resumen.
Las condiciones que debe satisfacer una función resumen son:

1. *Facilidad de cálculo.* Dado un mensaje m, el cálculo de $y = h(m)$ es sencillo y rápido.
2. *Unidireccionalidad.* Dado un resumen $h(m)$ es computacionalmente muy costoso encontrar m tal que $y = h(x)$.
3. *Difusión.* El resumen debe ser función de todos los bits del mensaje. Un cambio en un bit debe afectar a aproximadamente la mitad de los bits del resumen.
4. *Compresión.* El resumen debe tener longitud fija.
5. *Resistencia débil a colisiones.* Dado m es computacionalmente intratable encontrar $m' \neq m$ tal que $h(m) = h(m')$
6. *Resistencia fuerte a colisiones.* Es computacionalmente inviable encontrar m y m', distintos, tales que $h(m) = h(m')$.

Podría parecer que las condiciones tercera y cuarta son similares. Nada más lejos de la realidad. Para comprobarlo, veamos el siguiente ejemplo, conocido como *paradoja del cumpleaños.*

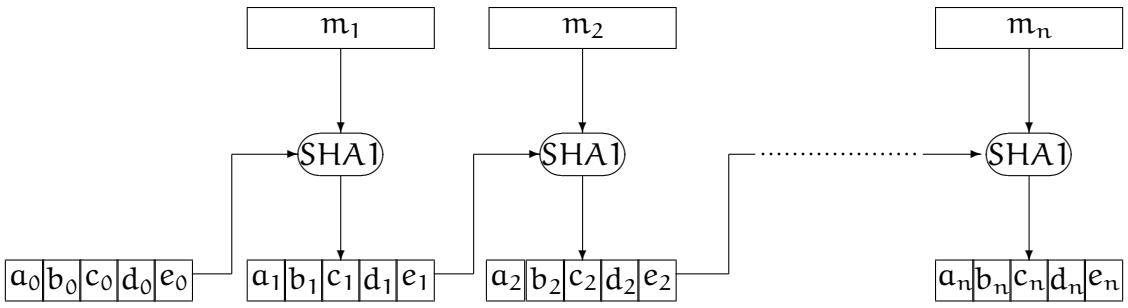
> Podemos considerar como conjunto de "mensajes" las personas, y como resumen la fecha de su cumpleaños (supongamos que nadie ha nacido el 29 de febrero).
>
> Tomamos a una persona P. ¿Cuál es la probabilidad de que, elegidas otras 50 personas haya alguna que celebre el cumpleaños el mismo día?.
>
> Para hallar esta probabilidad, hallamos la del suceso contrario, es decir, que ninguna celebre el cumpleaños el mismo día que P. Si fuera sólo una persona, la probabilidad sería $\frac{364}{365}$. Para la segunda, habría que multiplicar ésta probabilidad por $\frac{364}{365}$, y así sucesivamente hasta llegar a 50. Por tanto, la probabilidad de que haya alguna a la que le coincida el cumpleaños con P es
>
> $$1 - \frac{364^{50}}{365^{50}} \approx 1 - 0'8718 = 0'1281$$
>
> Supongamos ahora que tenemos 51 personas. ¿Cuál es la probabilidad de que haya dos que celebren el cumpleaños el mismo día?
>
> Elegimos una persona P_1. La probabilidad de que no haya coincidencias con sólo esa persona es $1 = \frac{365}{365}$. Si ahora tomamos una segunda persona P_2, la probabilidad de que no haya coincidencias es $\frac{364}{365}$. Al introducir una tercera, la probabilidad de que siga sin haber coincidencias es $\frac{364}{365} \cdot \frac{363}{365}$. Una vez consideradas las 51 personas, la probabilidad de que haya al menos alguna coincidencia es
>
> $$1 - \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdots \frac{315}{365} = 1 - \frac{365!}{365^{51} \cdot 314!} \approx 1 - 0'0255 = 0'9745$$
>
> Vemos como es mucho más fácil que falle la cuarta condición a la tercera, o lo que es lo mismo, es mucho más fácil que se cumpla la tercera a que se cumpla la cuarta. De ahí que a una se le llame *resistencia débil a colisiones.*

Son muchas las funciones Hash que han aparecido a lo largo de los años. Por citar algunas, tenemos MD2, MD4, MD5, SHA1, RIPEMD, Tiger, Panama, etc.
De ellas, probablemente la más usada en la actualidad ser la función SHA1. El nombre viene de las iniciales en inglés de Algoritmo Hash seguro (Secure Hash Algoritm). En realidad SHA constituye una familia de funciones Hash. La primera fue publicada por el NIST en 1993 (acualmente se la conoce como SHA0). En 1995 se publicón la SHA1. Esta función produce resúmenes de 160 bits y es similar al algoritmo MD5, aunque este produzca resúmenes de 128 bits. Hay otras propuestas, como SHA-256 o SHA-512.
Aunque se han encontrado colisiones a la función SHA1, estas aún no comprometen la seguridad del algoritmo, aunque podría verse comprometido en un futuro no muy lejano.
A continuación vamos a describir con detalle el algoritmo SHA1

. . . . . 3.2

Función SHA1

En primer lugar, notemos que el SHA1 no opera con mensajes de más de 2^{64} bits. Esta restricción no lo es tal, pues 2^{64} bits supone algo más de dos trillones de bytes.
El algoritmo opera con bloques de 512 bits. Por tanto, lo primero que hay que hacer es completar el mensaje hasta que el número de bits sea múltiplo de 512. Esto se hace de la siguiente manera.
En primer lugar, se añade un 1, seguido de tantos ceros hasta conseguir que le tamaño sea congruente con 448 módulo 512. Los 64 bits restantes que se añaden represetan la longitud del mensaje original. Una vez hecho esto, se divide el mensaje en bloques de 512 bits. Supongamos que estos bloques son $m_1, m_2, \cdots, m_n$.
El algoritmo trabaja con palabras de 32 bits. Por tanto, cada bloque se divide en 16 palabras (comenzando de izquierda a derecha).
Un esquema de como funciona SHA1 podría ser:



Se parte de un vector inicial de 160 bits (es decir, 5 palabras), y eso, junto con el primer bloque del mensaje da como salida otro vector de 160 bits. Este sirve de entrada, junto con el segundo bloque al algoritmo SHA1, y devuelve otro vector de 160 bits, y así hasta el último bloque.
El algoritmo SHA1 tiene 80 rondas. En cada ronda, el vector de 160 bits se va transformando. Tras la octagésima transformación se le suma el vector antes de la primera transformación, y esa es la salida.

Vamos a ver en detalle cómo se hacen todas estas transformaciones.
Tenemos como entrada un vector formado por cinco palabras $(a_i, b_i, c_i, d_i, e_i)$, cada una de ellas formada por 32 bits, y un bloque m_{i+1} de 512 bits, que dividimos en 16 palabras $(W_0, W_1, \cdots, W_{15})$. A partir de estas 16 palabras construimos 64 más hasta formas un total de 80. Cada una de ellas se empleará en cada una de las rondas. Las diferentes palabras se obtienen como sigue:

$$W_t = S^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) : \ 16 \leq t \leq 79$$

Se crea un vector de 5 palabras $(A_0, B_0, C_0, D_0, E_0)$ que inicialmente toma el valor $(a_i, b_i, c_i, d_i, e_i)$ este vector sufrirá distintas transformaciones:

$$(A_0, B_0, C_0, D_0, E_0) \to (A_1, B_1, C_1, D_1, E_1) \to (A_2, B_2, C_2, D_2, E_2) \to \cdots \to (A_{80}, B_{80}, C_{80}, D_{80}, E_{80})$$

Obtenido $(A_t, B_t, C_t, D_t, E_t)$ se construye $(A_{t+1}, B_{t+1}, C_{t+1}, D_{t+1}, E_{t+1})$ como sigue:

$$(A_{t+1}, B_{t+1}, C_{t+1}, D_{t+1}, E_{t+1}) = (g_t(A_t, B_t, C_t, D_t, E_t), A_t, S^{30}(B_t), C_t, D_t)$$

Por último, una vez terminadas las 80 rondas conseguimos el vector de salida como

$$(a_{i+1}, b_{i+1}, c_{i+1}, d_{i+1}, e_{i+1}) = (a_i + A_{80}, b_i + B_{80}, c_i + C_{80}, d_i + D_{80}, e_i + E_{80})$$

Todavía nos queda especificar algunos datos:

1. $a_0 = 67452301$, $b_0 =$ EFCDAB89, $c_0 =$ 98BZDCFE, $d_0 = 10325476$ y $e_0 =$ C3D2E1F0. Estos valores están expresados en hexadecimal.
2. $S^k(X)$ representa un desplazamiento circular hacia la derecha k posiciones de los bits de X. Por ejemplo:

$$S^5(10011101\ 10100011\ 11011110\ 00101001\ ) = 01001100\ 11101101\ 00011110\ 11110001$$

3. $g_t(X, Y, Z, T, U) = S^5(X) + F_t(Y, Z, T) + U + W_t + K_t$.
4. Las funciones F_t están definidas como sigue:

$$\begin{array}{ll} F_t(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z) & 0 \leq t \leq 19 \\ F_t(X,Y,Z) = (X \oplus Y) \oplus Z & 20 \leq t \leq 39 \\ F_t(X,Y,Z) = (X \wedge Y) \vee (Y \wedge Z) \vee (Z \wedge X) & 40 \leq t \leq 59 \\ F_t(X,Y,Z) = (X \oplus Y) \oplus Z & 60 \leq t \leq 79 \end{array}$$

5. Las palabras K_t son las siguientes:

$$\begin{array}{ll} K_t = 5A827999 & 0 \leq t \leq 19 \\ K_t = 6ED9EBA1 & 20 \leq t \leq 39 \\ K_t = 8F1BBCDC & 40 \leq t \leq 59 \\ K_t = CA62C1D6 & 60 \leq t \leq 79 \end{array}$$

6. $X + Y$ significa la suma de X e Y módulo 2^{32}.

4 HMAC

En los últimos años ha aumentado el interés por el desarrollo de un MAC basado en una función Hash, como la SHA1. El motivo es que las funciones Hash son mucho más rápidas de ejecutar que los algoritmos de cifrado, y que no hay restricciones para la exportación desde los Estados Unidos u otros países.
El que más apoyo ha recibido es el HMAC, que fue lanzado en 1996, y se emplea, por ejemplo, para la seguridad IP.
Los objetivos del diseño del HMAC son:

- Usa, sin modificaciones, las funciones resumen disponibles.
- Permite la sustitución de una función Hash por otra.
- Preserva el funcionamiento de una función Hash.
- Usa claves de forma sencilla.
- La robustez de este modo de autenticación está basado en la seguridad de la función Hash empleada.

Con esto, la implementación existente de una función Hash puede usarse como un módulo dentro de HMAC. Si se viera conveniente cambiar la función Hash porque se viera comprometida su seguridad, o porque queremos emplear otra más rápida no habría más que cambiar el módulo correspondiente.
El HMAC puede expresarse como sigue:

$$HMAC_K(m) = h[(K^+ \oplus opad)||h[(K^+ \oplus ipad)||m]]$$

Explicamos a continuación los distintos elementos que intervienen:

- h es la función resumen que se emplea.
- K es la clave secreta. Si la longitud de la clave es mayor que el tamaño de los bloques con los que opera h, se le hace el resumen.
- K^+ es el relleno, con ceros a la izquierda, de la clave K de forma que su tamaño coincida con el de los bloque con los que trabaja la función resumen.
- $ipad$ es el byte 36 (en hexadecimal) repetido tantas veces como sea necesario para tener un bloque del mismo tamaño que K^+.
- $opad$ es lo mismo que $ipad$ pero con $5C$.
- $\oplus$ es el "o exclusivo" o XOR.
- El símbolo $||$ denota "concatenación".

Vamos a ver los pasos que hemos de seguir para obtener el HMAC de un mensaje. Supondremos que la función resumen empleada es $SHA1$.

1. Calculamos K^+ a partir de K. En nuestro caso debemos rellenar con ceros hasta obtener 512 bits. Ya hemos visto que si K tiene más de 512 bits, hemos de calcular su resumen, que tendrá 160 bits, y a estos hay que añadirle 352 ceros.

2. Hacemos un "o exclusivo" de K^+ e $ipad$. Puesto que la mitad de los bits de $ipad$ son unos, lo que hacemos es cambiar la mitad de los bits de K^+.

3. Al bloque que nos ha dado (que tiene 512 bits) le unimos a la izquierda el mensaje del que queremos hallar su HMAC.

4. Al resultado del paso anterior le calculamos el resumen.

5. Hacemos un "o exclusivo" de K^+ y $opad$. Al igual que antes, lo que hacemos es invertir la mitad de los bits de K^+, aunque una mitad distinta.

6. Unimos las cadenas de bits obtenidas en los dos pasos anteriores.

7. Al resultado del paso 6 le aplicamos la función SHA1.

5 Firma digital RSA

Los dos ejemplos que hemos visto (MAC y HMAC) no son exactamente dos sistemas de firma digital. Bien es verdad que tenemos un espacio de mensajes, un espacio de firmas, un espacio de claves, una función de firma y una función de verificación. Pero sólo pueden verificar la firma quienes poseen la clave, que es secreta. Obviamente, si sirve para autenticar mensajes (comprobar su integridad).
Aunque hay varios sistemas de firma digital basados en criptografía simétrica, los más extendidos hacen uso de la criptografía asimétrica, pues ésta da muchas más posibilidades, y permite diseñar sistemas de firma sencillos.
En esta sección explicaremos el sistema de firma digital basado en el criptosistema RSA.
Supongamos que A tiene un mensaje que quiere firmar, y que A posee un cripsositema RSA de llave pública (n, e) y llave privada d. Una primera forma de firmar el mensaje es añadiéndole al mensaje el propio mensaje cifrado con la llave privada de A, es decir, la firma sería $fir(m) = m^d (\text{mód } n)$ (en el caso de un mensaje largo, habría que dividirlo en submensajes m_i de forma que $m = m_0 + m_1 \cdot n + \cdots + m_k \cdot n^k$).
El mensaje firmado sería el par $(m, fir(m))$.
Para comprobar si la firma es válida, habría que si $m = fir(m)^e (\text{mód } n)$.
Puesto que únicamente A posee la llave privada d, únicamente A es capaz de generar la firma. Por tanto, A no puede negar que ha sido él quien ha firmado el documento.
Además, cualquiera podría comprobar la validez de la firma, pues tanto e como n son públicos.

Pero este sistema, tal y como lo hemos descrito presenta un gran inconveniente. Para documentos largos, el proceso de firma y verificación es muy lento. La solución a esto se encuentra si en lugar de firmar el mensaje, firmamos un resumen del mismo. La firma quedaría $fir(m) = h(m)^d (\text{mód } n)$
La comprobación ahora requiere ver que $h(m) = fir(m)^e (\text{mód } n)$.
Si alguien quisiera modificar el documento, tendría que encotrar otro mensaje m' tal que $h(m) = h(m')$ y sustituir el par $(m, fir(m))$ por $(m', fir(m))$. Si la función h es una buena función resumen, sabemos que no es posible (o es casi imposible) encontrar ese mensaje m'.
Al igual que antes, sólo A puede generar esa firma, y cualquiera puede comprobarla, pues tanto n como e como h son públicos.

6

Firma ElGamal

Sea (p, α, y) la clave pública de un criptosistema ElGamal, cuya clave privada es x.
Sea m un mensaje que quiere firmar el poseedor de la clave privada. Para ello, elige un número aleatorio k tal que $2 \leq k \leq p-2$, primo relativo con $p-1$, y firmaría como sigue:

$$fir(m) = (r, s): \qquad \begin{array}{l} r = \alpha^k (\text{mód } p) \\ s = (m - x \cdot r) \cdot k^{-1} (\text{mód } p-1) \end{array}$$

Para calcular el valor de la firma, primero calculamos α^k y lo reducimos módulo p, y luego ese resultado es el que interviene en el cálculo de la segunda parte de la firma. Los cálculos, en este caso, hay que realizarlos módulo $p-1$.
Para comprobar la validez de la firma calculamos $y^r \cdot r^s (\text{mód } p)$ y $\alpha^m (\text{mód } p)$ Si ambos coinciden, la firma es válida.
Para ver porqué esto permite comprobar si la firma es válida o no, hemos de tener en cuenta que $\alpha^{p-1} \equiv 1 (\text{mód } p)$, luego para cualesquiera a, b tales que $a \equiv b (\text{mód } p-1)$ se verifica que $\alpha^a \equiv \alpha^b (\text{mód } p)$. Con esto, se tiene que (trabajando módulo p):

$$y^r \cdot r^s = (\alpha^x)^r \cdot (\alpha^k)^{(m - x \cdot r) \cdot k^{-1}} = \alpha^{x \cdot r} \cdot \alpha^{(k \cdot k^{-1}) \cdot (m - x \cdot r)} = \alpha^{x \cdot r} \cdot \alpha^m \cdot \alpha^{-x \cdot r} = \alpha^m$$

Ejemplo:
Veamos un ejemplo. Tomamos un criptosistema ElGamal con clave pública $(181081, 2757, 161952)$ y clave privada 97825.
Tomamos el mensaje $m = 13579$ para firmarlo, y elegimos $k = 142573$.

- $r = 2757^{142573} (\text{mód } 181081) = 29469$
- $142573^{-1} (\text{mód } 181080) = 142477$.
- $s = 142477 \cdot (13579 - 97825 \cdot 29469) (\text{mód } 181080) = 88318$.

Luego la firma es $(29469, 88318)$.
Para comprobar la firma, procedemos como sigue:

- $2757^{13579} (\text{mód } 181081) = 134234$.
- $142573^{29469} \cdot 29469^{88318} (\text{mód } 181081) = 9923 \cdot 56347 (\text{mód } 181081) = 134234$.

Luego la firma es válida.
Si para el cálculo de s hubiéramos hecho $s = m - x \cdot \alpha^k) \cdot k^{-1} (\text{mód } p-1)$ tendríamos:

$$s = (13579 - 97825 \cdot 2757^{142753}) \cdot 142477 (\text{mód } 181080) = (13579 - 97825 \cdot 41157) \cdot 142477 = 165478$$

y a la hora de verificar la firma tendríamos que

$$y^r \cdot r^s (\text{mód } 181081) = 9923 \cdot 100225 (\text{mód } 181081) = 35823$$

que no coincide con $\alpha^m (\text{mód } p)$.
El problema con el sistema de firma de ElGamal consiste en probar que es seguro, en el sentido de que un observador no puede producir una firma para un mensaje x a no ser que conozca la llave secreta a.
En efecto, si se considera un mensaje x dado, para obtener una firma para x tenemos que determinar r y s verificando

$$h^r r^s \equiv g^x \pmod{p}$$

tomando el logaritmo discreto se verifica:

$$x = \log_g(h^r r^s)$$

y determinar r o s exige resolver el problema del logaritmo discreto.
Este sistema de firma tiene un inconveniente. Sin conocer la clave privada podríamos obtener mensajes junto con su firma. Veamos cómo hacerlo.

- Elegimos enteros i, j tales que $0 \leq i, j \leq p-2$ y $mcd(j, p-1) = 1$.
- Hacemos $r = \alpha^i \cdot y^j (\text{mód } p)$
- Hacemos $s = -r \cdot j^{-1} (\text{mód } p-1)$
- Hacemos $m = -r \cdot i \cdot j^{-1} (\text{mód } p-1)$

En tal caso, podríamos enviar (m, r, s) como un mensaje firmado por el poseedor del criptosistema, pues:

$$y^r \cdot r^s = \alpha^{x \cdot r} \cdot (\alpha^i \cdot y^j)^{-r \cdot j^{-1}} = \alpha^{x \cdot r} \cdot \alpha^{-r \cdot i \cdot j^{-1}} \cdot (\alpha^{x \cdot j})^{-r \cdot j^{-1}} = \alpha^{x \cdot r} \cdot \alpha^m \cdot \alpha^{-x \cdot r} = \alpha^m$$

(todos los cálculos se han realizado módulo p).
También se pueden obtener mensajes con firmas a partir del conocimiento de una firma válida. Supongamos que (m, r, s) es una firma válida. Entonces podemos obtener otros mensajes junto con sus correspondientes firmas:
Para esto, procedemos como sigue:

- Elegimos enteros i, j, l tales que $0 \leq i, j, l \leq p-2$ y tales que $mcd(l \cdot r - j \cdot s, p-1) = 1$.
- Hacemos $r' = r^l \cdot \alpha^j \cdot y^i (\text{mód } p)$.
- Hacemos $s' = s \cdot (l \cdot r - j \cdot s)^{-1} (\text{mód } p-1)$.
- Hacemos $m' = r' \cdot (l \cdot m + i \cdot s) \cdot (l \cdot r - j \cdot s)^{-1} (\text{mód } p-1)$.

Y con esto conseguimos que la firma (r', s') sea válida para el mensaje m'.
Estos problemas pueden solucionarse si en lugar de firmar el mensaje, se firma el resumen del mensaje. En tal caso, una vez obtenido m' cuya firma es (r', s') habría que encontrar un mensaje m'' tal que $h(m'') = m'$. Si la función resumen es unidireccional, esto es prácticamente imposible de conseguir.
Una precaución que hay que tener con la firma ElGamal es no firmar dos mensajes (o dos resúmenes) con el mismo número aleatorio k.
Supongamos que se hiciera así. Sean (m, r, s) y (m', r, s') dos mensajes que se han firmado usando el mismo valor de k. En tal caso, tal vez pueda recuperarse la clave privada. Para eso, tenemos en cuenta que en $\mathbb{Z}_{p-1}$ se tienen las igualdades:

$$k \cdot s = m - x \cdot r \qquad k \cdot s' = m' - x \cdot r$$

Multiplicando la primera por s' y la segunda por s obtenemos:

$$\left.\begin{array}{l} k \cdot s \cdot s' = m \cdot s' - x \cdot r \cdot s' \\ k \cdot s \cdot s' = m' \cdot s - x \cdot r \cdot s \end{array}\right\} \Longrightarrow m \cdot s' - x \cdot r \cdot s' = m' \cdot s - x \cdot r \cdot s$$

Y de ahí tenemos la siguiente congruencia:

$$x \cdot r \cdot (s' - s) \equiv m \cdot s' - m' \cdot s (\text{mód } p-1)$$

Si $mcd(r \cdot (s'-s), p-1) = 1$ esta congruencia tiene solución única, luego tendríamos el valor de x. Pero aún si el máximo común divisor no es 1. Pero aún sin que ese máximo común divisor valga 1, esta congruencia tiene solución, y el número de soluciones de la congruencia menores que $p-1$ es $mcd(r \cdot (s'-s), p-1) = 1$. Calculándolas, podemos calcular x.
Ejemplo:
Supongamos que con el mismo criptosistema anterior firmamos el mensaje $m' = 34567$, usando el mismo valor k de antes (142573). La firma del mensaje sale $(r, s') = (29469, 40474)$.
Entonces:
$m \cdot s' - m' \cdot s = 139140$ y $r \cdot (s' - s) = 155124$.
Por tanto, planteamos la congruencia:

$$155124 \cdot x \equiv 139140 (\text{mód } 181080)$$

Puesto que $mcd(155124, 181080) = 36$ dividimos toda la congruencia por 36, y nos queda:

$$4309 \cdot x \equiv 3865 (\text{mód } 5030)$$

Y ahora, 4309 tiene inverso módulo 5030, y vale 1479. Y puesto que $3865 \cdot 1479 (\text{mód } 5030) = 2255$, la congruencia anterior es equivalente a:

$$x \equiv 2255 (\text{mód } 5030)$$

cuyas soluciones son:

$$x = 2255 + 5030 \cdot l : \ l \in \mathbb{Z}$$

En total hay 36 soluciones menores que 181081, y éstas son:

$$\left\{ \begin{array}{rrrrrrrrr} 2255 & 7285 & 12315 & 17345 & 22375 & 27405 & 32435 & 37465 & 42495 \\ 47525 & 52555 & 57585 & 62615 & 67645 & 72675 & 77705 & 82735 & 87765 \\ 92795 & 97825 & 102855 & 107855 & 112915 & 117945 & 122975 & 128005 & 133035 \\ 138065 & 143095 & 148125 & 153155 & 158185 & 163215 & 168254 & 173275 & 178395 \end{array} \right\}$$

Y vemos que la vigésima (la que se obtiene para $k = 19$) se corresponde con el valor de x.
Para calcular el valor de x habría que ir elevando α a las distintas soluciones hasta obtener y.

..... 7

DSS

Éste es el estándar de firma digital propuesto por el NIST, y se corresponde con las siglas de *Digital Signature Standard*. Este algoritmo es una modificación de la firma ElGamal que acabamos de estudiar.
Los parámetros necesarios para este algoritmo de firma son:

- Un primo p comprendido entre 2^{L-1} y 2^L, donde L es un número natural sujeto a las restricciones $512 \leq L \leq 1024$.
- Un primo q, comprendido entre 2^{159} y 2^{160} (es decir, un primo de 160 bits).
- Un elemento α de orden q en $\mathbb{Z}_p$, es decir, un elemento $\alpha \in \mathbb{Z}_p$ para el que $\alpha^q \equiv 1 (\text{mód } p)$ y q es el primer exponente para el que ocurre eso (para esta última condición basta con que $\alpha \neq 1$).
- Un entero x tal que $2 \leq x \leq q - 2$.
- El entero y, menor que p dado por la condición $\alpha^x \equiv y (\text{mód } p)$.

Los valores p, q, α, y son públicos, mientras que x permanece secreto.
A continuación vamos a ver como podemos elegir los parámetros p, q, α.

1. En primer lugar, hallamos q. Éste debe ser un número primo de 160 bits. Elegimos al azar un número impar de 160 bits

2. Comprobamos si q es primo (test de Miller-Rabin). Si no lo es, le sumamos a q dos y volvemos a probar, hasta que obtengamos un valor que sea primo.
3. Ahora hay que elegir un primo p de L bits, tal que $p \equiv 1 (\text{mód } q)$. Es decir, $p = c \cdot q + 1$ para algún entero c. El tamaño de c debe ser $L - 160$ bits, y debe ser par. Por tanto, elegimos c al azar con esas condiciones y probamos si $p = c \cdot q + 1$ es o no primo.
4. Si p no es primo, le sumamos a p $2q$ (o lo que es lo mismo, le sumamos 2 a c), y volvemos a probar. Así hasta que obtengamos un número primo.
5. Por último, nos queda elegir un elemento que tenga orden q. Para esto elegimos al azar un número $2 \leq g \leq p - 2$. Calculamos $\alpha = g^{\frac{p-1}{q}}$ módulo p. Si $\alpha \neq 1$ ya tenemos g. Si $\alpha = 1$ debemos repetir el proceso con otro α (podría ser $\alpha + 1$).

Para firmar un mensaje m procedemos como sigue:

1. Se calcula un resumen del mismo $h(m)$ con una función resumen (por ejemplo, SHA-1)
2. Se elige un número aleatorio k tal que $2 \leq k \leq q - 2$.
3. Se calcula $r = [\alpha^k \ (\text{mód } p)](\text{mód } q)$.
4. Se calcula $s = (h(m) + x \cdot r) \cdot k^{-1}$ en $\mathbb{Z}_q$.

Se tiene entonces que $fir(m) = (r, s)$.
Por último, para comprobar la firma supongamos que tenemos un mensaje m junto con su firma (r, s).
Hacemos los siguientes cálculos en $\mathbb{Z}_q$:

1. Calculamos en $\mathbb{Z}_q$ $u = h(m) \cdot s^{-1}$
2. Calculamos, también en $\mathbb{Z}_q$, $v = r \cdot s^{-1}$.
3. Calculamos $r' = [\alpha^u \cdot y^v \ (\text{mód } p)](\text{mód } q)$.

Si $r = r'$ la firma es válida.
Si $r \neq r'$ la firma no es válida.
Efectivamente, si la firma es válida, $s = (h(m)+x\cdot r)\cdot k^{-1}$ en $\mathbb{Z}_q$, luego $k \equiv s^{-1} \cdot (h(m) + x \cdot r)(\text{mód } q)$, y puesto que α es un elemento de orden q en $\mathbb{Z}_p$, entonces $\alpha^k = \alpha^{s^{-1}\cdot(h(m)+x\cdot r)}$ en $\mathbb{Z}_p$.
Vamos a calcular $\alpha^u \cdot y^v \ (\text{mód } p)$:

$$\alpha^u = \alpha^{h(m)\cdot s^{-1}}, \qquad y^v = (\alpha^x)^v = \alpha^{x\cdot r\cdot s^{-1}},$$

luego:

$$\alpha^u \cdot y^v = \alpha^{h(m)\cdot s^{-1}} \cdot \alpha^{x\cdot r\cdot s^{-1}} = \alpha^{s^{-1}\cdot(h(m)+x\cdot r)} = \alpha^k$$

Y por tanto, $r' = [\alpha^k \ (\text{mód } p)](\text{mód } q) = r$.

8 Certificados digitales.

Imaginémonos una situación cotidiana. Una persona, en un comercio desea pagar sus compras con su tarjeta de crédito. Para eso, debe firmar un documento en el que da su conformidad a que se le descuente de su cuenta la cantidad que ha gastado. ¿Cómo puede estar seguro el responsable del comercio de que el que firma es el propietario de la tarjeta?. Normalmente, lo que se hace es pedir el DNI y con él comprobar la identidad del firmante: comparar el nombre que aparece en el DNI con el del propietario de la tarjeta, la foto del DNI con la apariencia de quien está realizando la compra, y la firma del DNI con la estampada en el documento.
Para poder comprobar la identidad de alguien en quien no se confíam (en este caso porque no se conoce) se recurre a un documento que da fe de que esa persona es quien dice ser. En este documento sí confiamos, pues confiamos en la entidad que ha emitido el documento (la Policía Nacional).
Es decir, para comprobar una identidad (autenticar a una persona) ha habido que recurrir a una tercera parte, en quien se confía, y que atestigua que la persona no está mintiendo acerca de su identidad.
A poco que lo pensemos situaciones en las que es necesario comprobar la identidad se presentan constantemente.
Para autenticar a una persona, únicamente necesitamos a alguien en quien confiemos que nos confirme esa identidad. Ese alguien puede ser el propio interesado, un amigo común, o alguna entidad como un banco, la Universidad, o el Estado.
En el mundo de las comunicaciones digitales la situación es muy similar. Si alguien firma un documento con una firma digital, ¿cómo puede convencer a quien lo reciba que la firma la ha realizado él?. Él puede comprobar que esa firma se ha realizado con una determinada clave privada, pero ¿corresponde esa clave privada con quien dice realizar la firma?. Ya vimos, cuando hablamos del protocolo de intercambio de Diffie-Hellman que un atacante podría situarse en medio de los dos comunicantes y convencer a los dos que su clave es la del otro comunicante. Es decir, si A y B quieren comunicarse, un intruso C podría interceptar las comunicaciones y convencer a A que la clave pública de B es la suya (la de C), y a B que la clave pública de A es también la suya.
Es aquí donde interviene lo que se llama una *tercera parte de confianza*, que como ya hemos dicho, es algo o alguien en quien confían tanto A como B. Claro, que para que tenga validez legal debe ser alguna entidad reconocida para ello.
Esta tercera parte de confianza, emite un *certificado digital* (también llamado certificado electrónico) que no es más que un documento digital en el que se relaciona una identidad con unos datos, y que va firmado digitalmente por quien lo emite.
La ley 59/2003, que ya hemos mencionado anteriormente declara:

- *Se denomina prestador de servicios de certificación la persona física o jurídica que expide certificados electrónicos o presta otros servicios en relación con la firma electrónica* (Artículo 2.2).

- *Un certificado electrónico es un documento firmado electrónicamente por un prestador de servicios de certificación que vincula unos datos de verificación de firma a un firmante y confirma su identidad* (Artículo 6.1).
- *Son certificados reconocidos los certificados electrónicos expedidos por un prestador de servidios de certificación que cumpla los requisitos establecidos en esta ley en cuanto a la comprobación de la identidad y demás circunstancias de los solicitantes y a la fiabilidad y las garantías de los servicios de certificación que presten* (Artículo 11.1).

Los prestadores de servicios de certificación se denominan también *Autoridades de Certificación.*
Todo certificado digital debe constar al menos de las siguientes partes:

1. Información relativa al propietario del certificado.
2. Información relativa al emisor del certificado.
3. Claves del propietario.
4. Firma digital del certificado por el emisor.

El formato X.509 es el más común de los certificados digitales. Un certificado X.509 es un conjunto estándar de campos con diversas informaciones referentes al usuario y al emisor. Actualmente, hay 3 versiones del certificado X-509. La primerea de 1988, la segunda de 1993 y la tercera de 1996.
Los certificados X.509 deben contener los siguientes datos:

Versión
Número de serie
Algoritmo de firma
Entidad emisora
Periodo de validez
Identificación del sujeto
Llave pública del sujeto
Identificador único de la entidad emisora
Identificador único del sujeto
Extensiones
Firma digital de la entidad emisora

Explicamos a continuación estos campos:

Versión: Indica la versión del estándar X.509 que se ha empleado. Puede ser la 1, la 2 o la 3.

Número de serie: Cada vez que una entidad crea un certificado debe asociarle un número de serie, que no puede repetir en ningún otro certificado.

Identificador del algoritmo de firma: Especifica el algoritmo que ha usado el emisor para firmar el certificado.

Entidad emisora: Nombre de la entidad que emite el certificado.

Periodo de validez: Indica el instante en que el certificado comienza a ser válido, y cuando dejará de estarlo.

Identificación del sujeto: Datos identificativos del poseedor del certificado.

Llave pública del sujeto: La llave pública del poseedor del certificado, como un identificador del criptosistema utilizado.

Identificador único de la entidad emisora: Una cadena de bits que se usa para identificar a la entidad emisora.

Identificador único del sujeto: Igual que el anterior, pero para el usuario.

Firma digital de la entidad emisora: Firma de todos los campos anteriores del certificado realizada con la llave privada de la entidad emisora.

En España la entidad que más certificados digitales ha emitido es la Fabrica Nacional de Moneda y Timbre (FNMT), a través de la autoridad pública de Certificación española (CERES). Para solicitar un certificado digital a la Fábrica Nacional de Moneda y Timbre hemos de entrar en la página de internet de CERES (http://www.cert.fnmt.es/). Una vez allí hemos de rellenar la solicitud, para lo cual nos pide que introduzcamos nuestro NIF. Una vez enviada la solicitud, en nuestro ordenador se generan nuestras claves, pues CERES necesita nuestra clave pública para emitir el certificado.
En el proceso se nos pide que indiquemos que grado de seguridad queremos. Es conveniente pedirlo alto, pues de esta forma sólo se podrá usar el certificado tras introducir una contraseña. De no ser así, cualquiera que acceda a un ordenador donde hayamos instalado el certificado, suplantando nuestra identidad.
Una vez generadas las claves, ambas (la privada y la pública) quedan almacenadas en nuestro ordenador, mientras que la pública es enviada por el navegador a la FNMT. Una vez recibida, la FNMT nos devuelve un código que necesitaremos posteriormente.
Terminado este proceso, deberemos presentarnos físicamente en alguna de las oficinas de registro locales (en la página de CERES podemos consultar a donde podemos dirigirnos, aunque la información no está, de momento, actualizada). En ellas, tras ser identificados, deberemos firmar de forma manuscrita la solicitud de certificado. En este instante deberemos dar una dirección de correo electrónico.
A los pocos días, recibiremos un correo electrónico de la FNMT donde se nos informa que ya podemos descargarnos el certificado. Esto habrá de hacerse en el mismo ordenador con

el que se hizo la solicitud. Deberemos entrar en una página e identificarnos. Entonces, se descargará el certificado en nuestro ordenador, y se instalará automáticamente. Si utilizamos el navegador Explorer, podremos ver el certificado entrando en

`Herramientas` → `Opciones de Internet` → `Contenido` → `Certificados` → `Personal`

y eligiendo el certificado que lleva nuestro nombre. Si en lugar de `Personal` elegimos alguna otra pestaña, podremos ver los distintos certificados que tenemos instalados en nuestro ordenador.

CRIPTOGRAFÍA XIII

Protocolos criptográficos

1

Introducción

Un *protocolo* es una secuencia ordenada de pasos realizados por dos o más partes para realizar alguna tarea. Todos aquellos usuarios del protocolo deben conocerlo y estar de acuerdo con él antes de usarlo. Tan importante es el orden de los pasos como la actividad de cada uno. Todo protocolo tiene las siguientes características:

- *Previamente establecido:* El diseño del protocolo está completamente terminado antes de su uso.
- *Mutuamente suscrito:* Todos los participantes en el protocolo están de acuerdo en seguir sus pasos en orden.
- *Sin ambigüedad:* Nadie puede fallar al dar un paso por haberlo entendido incorrectamente.
- *Completo:* Hay una acción prescrita para cada situación que pueda ocurrir.

Podemos destacar dos motivos para desarrollar el uso de protocolos. Debido a la distancia, y el anonimato que conlleva, que se presenta hoy en día en las comunicaciones entre sistemas informáticos, los usuarios de éstos pueden no confiar en los administradores u otros usuarios de sistemas con los que van a establecer comunicación (de hecho no deben confiar en los mismos). El establecimiento de un protocolo permite interactuar a usuarios de sistemas distintos con el convencimiento de que no van a resultar engañados. Por otra parte, la definición de un protocolo permite diferenciar el proceso seguido para realizar una tarea del mecanismo mediante el cual se realiza. Una vez convencidos de la corrección del diseño del protocolo podemos implementarlo mediante algún mecanismo, ya sea un lenguaje concreto o un criptosistema determinado. De esta forma sólo tenemos que verificar que el mecanismo refleja fielmente el diseño, y no comprobar que la implementación resuelve el problema para el que se diseñó el protocolo.
Recordemos que la criptografía actual pretende resolver diversos problemas de seguridad que se pueden platear en las comunicaciones digitales. En todos estos problemas siempre hay

algún dato (una clave, un número de cuenta, la combinación de una caja fuerte ...) que hay que mantener secreto, o sólo al alcance de cierto personal autorizado. En este contexto podemos plantearnos multitud de preguntas:

1. Si el dato secreto lo obtengo a través de un distribuidor, ¿podría él acceder a esta información sin mi consentimiento?

2. Si yo faltara por algún motivo, ¿quién podría recuperar esa información?. Quizá podría entregar duplicados a ciertas personas, pero ¿puedo fiarme de todas ellas?.

3. También podría repartir la información en varias participaciones, pero ¿y si, llegado el momento, alguien se niega a aportar su parte?.

Estas y otras muchas cuestiones pueden surgirnos cuando deseamos realizar alguna acción a través de Internet. La criptografía moderna tiene respuestas a estas situaciones, a través de unos protocolos que detallan una serie de acciones concretas que deben realizar los interlocutores en el orden establecido. En lo que resta de capítulo estuidaremos algunos de ellos:

2 Protocolo del lanzamiento de una moneda.

Comenzamos con un protocolo muy sencillo. La idea es simular el lanzamiento de una moneda entre dos personas que están comunicadas a través de la red.
Cuando las dos personas están presentes en el lanzamiento de la moneda, uno hace una apuesta (cara o cruz) y el otro (o él mismo) lanza la moneda, y cada uno tiene probabilidad $\frac{1}{2}$ de ganar la apuesta. Al estar los dos juntos, ninguno puede engañar al otro, pues al lanzarse la moneda, los dos ven simultáneamente el resultado. Pero si el único contacto que tienen es a través de internet, el que lanza la moneda, que ya conoce la apuesta, podría engañar a su interlocutor.
El siguiente protocolo trata de evitar que se produzca este engaño. Su fundamento se encuentra en el apartado dedicado a *raíz cuadrada y factorización.*
Supongamos que A y B desean simular el lanzamiento de una moneda, y su única forma de comunicarse es a través de Internet (o a través del teléfono). Entonces pueden proceder como sigue:

1. A elige dos números primos grandes p y q. Calcula su producto y se lo envía a B.

2. B elige un número al azar u entre 1 y $\frac{n-1}{2}$, calcula $v = u^2 (\text{mód } n)$ y lo envía a A.

3. A calcula las cuatro raíces cuadradas de v módulo n, lo cual es posible pues A conoce la factorización de n. Elige una que sea menor que $\frac{n}{2}$ (hay dos en esa situación) y la envía a B. Supongamos que la raíz que ha elegido es z.

Si A ha elegido la raíz u, entonces gana la apuesta, pero si ha elegido la otra raíz entonces es B quien gana la apuesta. En caso de que sea B quien gana, puede demostrarlo factorizando n, mientras que si es A quien la gana, entonces B no puede encontrar los factores primos de n.
Como A tiene que elegir una raíz entre dos posibles, tiene una probabilidad del 50 % de ganar, la misma que tiene B.

3 Protocolos de secreto compartido

Pongámonos en la situación de una entidad que debe repartir una clave secreta entre los distintos miembros de la organización. Para recuperar la clave se requiere la participación de un número determinado de miembros. Así, si alguien quisiera obtener la clave, él sólo no podrá hacerlo, sino que deberá convencer a otros miembros para ello. Y si fuera necesario recuperarla, si algunos se niegan, el resto pueden recuperarla.
Vamos a describir un procedimiento para ésto: el método umbral de Shamir.
La idea es muy sencilla. Todos sabemos que una recta está determinada por dos puntos. Se podría entonces "esconder" la información en una recta, de forma que conocida la recta, se puede recuperar esa información. Entonces a cada participante se le da un punto de la recta. Sería necesario entonces que se juntaran dos participantes, para, con sus dos puntos, poder calcular la recta.
Si en lugar de la recta, cuya ecuación viene dada por un polinomio de grado 1, utilizamos polinomios de grado mayo, entonces necesitaremos un mayor número de puntos para recuperar la información.
El procedimiento que vamos a describir pretende distribuir un secreto s en n participaciones, de forma que sean necesarias t de ellas para poder recuperarlo. Es lo que se llama un método umbral de parámetros (n, t). Para poner en práctica este sistema de distribución:

- Se elige un número primo p que sea mayor que s y mayor que n.
- Se eligen $t-1$ elementos de $\mathbb{Z}_p$, $a_1, a_2, \cdots, a_{t-1}$, con $a_{t-1} \neq 0$.
- Se toma el polinomio $P(x) = s + a_1x + \cdots + a_{t-1}x^{t-1}$.
- Se eligen n elementos distintos $x_1, x_2, \cdots, x_n$ en $\mathbb{Z}_p^*$.
- Se calcula $y_1 = P(x_1), y_2 = P(x_2), \cdots, y_t = P(x_n)$.
- A cada participante se le da la pareja de números (x_i, y_i).

Vamos a ver ahora cómo se recupera el secreto. En primer lugar, recordemos que dados t puntos del plano, de abscisas diferentes, hay un único polinomio de grado menor o igual que $t-1$ que pasa por esos puntos. Si tuviéramos, en lugar de t puntos k puntos con $k < t$, entonces el número de polinomios con coeficientes en $\mathbb{Z}_p$ que pasan por esos puntos es p^{t-k}. Por tanto, con menos de t puntos es imposible determinar el polinomio.

Supongamos que se juntan los participantes $i_1, i_2, \cdots i_t$. Para recuperar el secreto, hay que encontrar el polinomio interpolador para los puntos $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \cdots, (x_{i_t}, y_{i_t})$. Para simplificar la notación supondremos que los t participantes son los t primeros.
Tenemos dos métodos para calcular el polinomio (bueno, hay más). El método de Lagrange y el método de Newton.

Método interpolador de Lagrange

Para encontrar el polinomio, se calculan previamente los polinomios interpoladores de Lagrange:

$$\begin{aligned} P_1(x) &= \frac{(x-x_2)\cdot(x-x_3)\cdots(x-x_t)}{(x_1-x_2)\cdot(x_1-x_3)\cdots(x_1-x_t)} \\ P_2(x) &= \frac{(x-x_1)\cdot(x-x_3)\cdots(x-x_t)}{(x_2-x_1)\cdot(x_2-x_3)\cdots(x_2-x_t)} \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ P_t(x) &= \frac{(x-x_1)\cdot(x-x_2)\cdots(x-x_{t-1})}{(x_t-x_1)\cdot(x_t-x_2)\cdots(x_t-x_{t-1})} \end{aligned}$$

Y a partir de estos polinomios, el polinomio solución es:

$$P(x) = y_1 \cdot P_1(x) + y_2 \cdot P_2(x) + \cdots + y_t \cdot P_t(x)$$

En la siguiente tabla, vamos a escribir en las columnas de la derecha, los polinomios que tenemos en la columna de la izquierda evaluados en diferentes puntos. Así se verá porqué este polinomio interpola los valores dados.

	x_1	x_2	.....	x_t
$P_1(x)$	1	0	.....	0
$P_2(x)$	0	1	.....	0
.....	..	..	.....	..
$P_t(x)$	0	0	.....	1
$P(x)$	y_1	y_2	.....	y_t

Método de interpolación de Newton

El siguiente algoritmo nos da el polinomio deseado.

$P(x) := y_1$

$Q(x) := x - x_1$

i:=2

Mientras $i \leq t$

$P(x) := Q(x_i)^{-1} \cdot (y_i - P(x_i)) \cdot Q(x) + P(x)$

$Q(x) := Q(x) \cdot (x - x_i)$

$i := i + 1$

El valor final de $P(x)$ es el polinomio interpolador.
En realidad, ambos métodos lo que hacen es resolver el siguiente sistema de congruencias en $\mathbb{Z}_p[x]$

$$\begin{aligned} P(x) &\equiv y_1 (\text{mód } x - x_1) \\ P(x) &\equiv y_2 (\text{mód } x - x_2) \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ P(x) &\equiv y_t (\text{mód } x - x_t) \end{aligned}$$

El método de Lagrange sigue los pasos de la demostración del teorema chino del resto, mientras que el método de Newton va resolviendo cada congruencia e introduciendo la solución en la siguiente, hasta llegar a la última.
Una vez obtenido el polinomio interpolador, el secreto se recupera tomando el término independiente (o calculando el valor $P(0)$.
Ejemplo:
Supongamos que tenemos un secreto $S = 32$, que queremos repartirlo entre 8 miembros de una empresa, de forma que sean necesarios 4 de ellos para recuperarlo.
Tomamos el primo $p = 37$, y los elementos de $\mathbb{Z}_{37}$ 12, 25 y 16.
Formamos el polinomio $P(x) = 32 + 12x + 25x^2 + 16x^3$.
Elegimos 8 elementos de $\mathbb{Z}_{37}$, por ejemplo, $5, 11, 19, 21, 25, 28, 30, 34$.
Evaluamos $P(x)$ en esos 8 puntos:

$$\begin{array}{llll} P(5) = 16 & P(11) = 28 & P(19) = 0 & P(21) = 15 \\ P(25) = 1 & P(28) = 16 & P(30) = 14 & P(34) = 11 \end{array}$$

Tenemos entonces 8 particiones del secreto que repartimos entre los 8 miembros. Éstas son:

$$(5,16) \quad (11,28) \quad (19,0) \quad (21,15) \quad (25,1) \quad (28,16) \quad (30,14) \quad (34,11)$$

Supongamos que se juntan los que poseen las participaciones $(5,33)$, $(19,6)$, $(25,1)$ y $(34,34)$ para acceder al secreto. Entonces tendrían que calcular el polinomio. Hagámoslo primero por el método de Lagrange:

$$\begin{aligned} P_1(x) = \ & \tfrac{(x-19)\cdot(x-25)\cdot(x-34)}{(5-19)\cdot(5-25)\cdot(5-34)} = \tfrac{x^3-78x^2+1971x-16150}{-8120} = \tfrac{x^3+33x^2+10x+19}{20} = \\ & 13 \cdot (x^3 + 33x^2 + 10x + 19) = 13x^3 + 22x^2 + 19x + 25 \end{aligned}$$

$$\begin{aligned} P_2(x) = \ & \tfrac{(x-5)\cdot(x-25)\cdot(x-34)}{(19-5)\cdot(19-25)\cdot(19-34)} = \tfrac{x^3-64x^2+1145x-4250}{1260} = \tfrac{x^3+10x^2+35x+5}{2} = \\ & 19 \cdot (x^3 + 10x^2 + 35x + 5) = 19x^3 + 5x^2 + 36x + 21 \end{aligned}$$

$$\begin{aligned} P_3(x) = \ & \tfrac{(x-5)\cdot(x-19)\cdot(x-34)}{(25-5)\cdot(25-19)\cdot(25-34)} = \tfrac{x^3-58x^2+911x-3230}{-1080} = \tfrac{x^3+16x^2+23x+26}{30} = \\ & 21 \cdot (x^3 + 16x^2 + 23x + 26) = 21x^3 + 3x^2 + 2x + 28 \end{aligned}$$

$$\begin{aligned} P_4(x) = \ & \tfrac{(x-5)\cdot(x-19)\cdot(x-25)}{(34-5)\cdot(34-19)\cdot(34-25)} = \tfrac{x^3-49x^2+695x-2375}{3915} = \tfrac{x^3+25x^2+29x+30}{30} = \\ & 21 \cdot (x^3 + 25x^2 + 29x + 30) = 21x^3 + 7x^2 + 17x + 1 \end{aligned}$$

Y ahora calculamos el polinomio interpolador como

$$16 \cdot P_1(x) + 0 \cdot P_2(x) + P_3(x) + 11 \cdot P_4(x) = 16x^3 + 25x^2 + 12x + 32$$

Como lo que nos hace falta es $P(0) = 32$, podríamos reducir los cálculos:

$$P_1(0) = \frac{(-19)\cdot(-25)\cdot(-34)}{(5-19)\cdot(5-25)\cdot(5-34)} = \frac{-16150}{-8120} = \frac{19}{20} = 20^{-1} \cdot 19 = 13 \cdot 19 = 29$$

$$P_2(0) = \frac{(-5)\cdot(-25)\cdot(-34)}{(19-5)\cdot(19-25)\cdot(19-34)} = \frac{-4250}{1260} = \frac{5}{2} = 2^{-1} \cdot 5 = 19 \cdot 5 = 21$$

$$P_3(0) = \frac{(-5)\cdot(-19)\cdot(-34)}{(25-5)\cdot(25-19)\cdot(25-34)} = \frac{-3230}{-1080} = \frac{26}{30} = 30^{-1} \cdot 26 = 21 \cdot 26 = 28$$

$$P_4(0) = \frac{(-5)\cdot(-19)\cdot(-25)}{(34-5)\cdot(34-19)\cdot(34-25)} = \frac{-2375}{3915} = \frac{30}{30} = 1$$

Y así obtener S como

$$S = P(0) = 16 \cdot 29 + 0 \cdot 21 + 1 \cdot 28 + 11 \cdot 1 = 32$$

Si lo resolvemos usando el algoritmo de Newton, tendríamos:

i	x_i	y_i	$P(x_i)$	$y_i - P(x_i)$	$Q(x_i)$	$Q(x_i)^{-1}$	$P(x)$	$Q(x)$
	5	16					16	$x + 32$
2	19	0	16	21	14	8	$20x + 27$	$x^2 + 13x + 21$
3	25	1	9	29	9	33	$32x^2 + 29x + 33$	$x^3 + 25x^2 + 29x + 30$
4	34	11	12	36	30	21	$16x^3 + 25x^2 + 12x + 32$	$x^4 + 28x^3 + 30x^2 + 6x + 16$

Y como vemos da el mismo resultado.
Podemos comprobar como:

$P(x) = 16$ es el polinomio de grado cero que interpola a $(5, 16)$

$P(x) = 20x + 27$ interpola a $(5, 16)$ y $(19, 0)$

$P(x) = 32x^2 + 29x + 33$ interpola a $(5, 16)$, $(19, 0)$ y $(25, 1)$.

El esquema que acabamos de ver nos muestra una distribución uniforme de las participaciones. Todas tienen el mismo peso. Pero supongamos que deseamos realizar una distribución jerarquizada de las particiones, de forma que el número mínimo para recuperar el secreto dependa de quienes son los que se reúnen. En tal caso, se puede repartir a cada miembro una o más participaciones, dependiendo del rango que le demos.
Por ejemplo, supongamos que cierta información de la Universidad se quiere repartir entre profesores y estudiantes.
Para acceder a la información, el rector y los vicerrectores queremos que puedan hacerlo ellos solos, pero si son profesores es necesario al menos dos, mientras que si son estudiantes son

necesarios al menos cinco, a menos que los estudiantes se unan a algún profesor, en cuyo caso sólo es necesaria la participación de dos estudiantes.
En este caso, vemos que podemos esconder la información en un polinomio de grado 4. Y se le dan a los vicerrectores y al rector 5 participaciones, tres a los profesores y una a los estudiantes.
De esta forma, dos profesores juntan 6 participaciones y pueden obtener la información. Un profesor y dos estudiantes también pueden acceder a dicha información, al igual que 5 estudiantes.
Pero también hay otra forma de realizar una distribución jerarquizada. Por ejemplo, supongamos que tenemos l clases distintas de participaciones, y que para recuperar el secreto queremos que sean necesarias n_1 participaciones de la clase 1, n_2 participaciones de la clase 2, y así hasta n_l participaciones de la clase l.
Entonces, dividimos el secreto S en l secretos parciales $S_1, S_2, \cdots, S_l$, de forma que el secreto se pueda obtener en función de los secretos parciales ($S = f(S_1, S_2, \cdots, S_l)$ para alguna función f. Un ejemplo de función f podría ser la suma). Entonces, el secreto S_1 se reparte entre los participantes de la clase 1 escondiéndolo en un polinomio de grado $n_1 - 1$. Lo mismo hacemos con el resto de los secretos parciales.
Para poder recuperar el secreto es necesario conocer todos los secretos parciales, y para recuperar éstos son precisos el número de participantes de la clase correspondiente que hemos indicado antes.

4 Protocolos de conocimiento cero

Supongamos que A posee cierta información y quiere convencer a B de que posee esa información. Pero no quiere desvelar a B esa información que posee.
Al final del proceso, la probabilidad de que A convenza a B de que posee la información sin poseerla realmente es mínima, y por otra parte B no ha adquirido ninguna información.
La técnica que se utiliza es la del desafío-respuesta. B lanza una pregunta a A, de forma que si A posee la información que dice tener, entonces puede responder con seguridad, pero si no la posee, entonces puede acertar la respuesta con cierta probabilidad (normalmente $\frac{1}{2}$. De esta forma, si A responde correctamente n desafíos, la probabilidad de que no tenga la información es $\frac{1}{2^n}$.
Este tipo de protocolos se utilizan a la hora de identificarse. Para eso, debemos hacer ver que poseemos cierta información privada, pero esa información no debe salir de nosotros.
Existen multitud de protocolos de conocimiento cero. Vamos a explicar dos de ellos:

4.1 *Protocolo de Chaum, Evertse y Van de Graaf*

Este protocolo fue publicado en 1987. En este caso, A trata de convencer a B de que posee la solución a un problema de logaritmo discreto. Como esa solución puede ser la llave privada

de un criptosistema ElGammal, puede convencer a B que posee cierta información cifrada con dicho criptosistema.
Los parámetros del protocolo son p un primo grande, α un elemento primitivo de $\mathbb{Z}_p$ e $y \in \mathbb{Z}_p^*$. Estos valores son conocidos tanto por A como por B. A quiere convencer a B de que conoce el valor del logaritmo, en base α de y (llamémosle x a dicho logaritmo, que existe por ser α primitivo). Es decir, la información que conoce A sería la llave privada de un criptosistema ElGammal de llave pública (p, α, y).
El protocolo entonces sería el siguiente:

1. A elige un valor aleatorio r, calcula $\beta = \alpha^r (\text{mód } p)$ y envía este valor a B.
2. B elige un bit c y lo envía a A.
3. A calcula $z = r + cx (\text{mód } p - 1)$.
4. B acepta la prueba si $\alpha^z = \beta \cdot y^c (\text{mód } p)$.

Obviamente, si $z = r + cx$ entonces $\alpha^z = \alpha^r \cdot \alpha^{xc} = \beta \cdot y^c$, luego si A conoce x puede superar el desafío.
Si A no conociera x podría intentar engañar a B. En el paso 1 A tiene dos opciones:

1. A envía a B el valor β que ha calculado.
2. A envía a B el valor $\beta' = \beta \cdot y^{-1}$.

Si A elige la opción 1, y B elige el bit $c = 0$, entonces A puede responder satisfactoriamente al desafío, pues no tiene más que enviar a B el valor r. Pero si B elige el bit $c = 1$, entonces A no tiene forma de superar el desafía.
Si A elige la opción 2, y B elige el bit $c = 0$, entonces A no puede responder satisfactoriamente. Si B eligiera el bit $c = 1$, entonces sí podría superar el desafío enviándole a B el valor r.
Por tanto, tanto si A elige la opción 1, como si elige la opción 2, tiene una probabilidad del 50 % de responder correctamente.

. . . . . 4.2

Protocolo de Fiat-Shamir.

Aquí A tiene un número de identificación ID, y su raíz cuadrada módulo n, donde n es producto de dos números primos. Quiere convencer a B de que posee esa raíz cuadrada, que denominaremos s. B conoce n, y A le envía a B el número ID.
El protocolo es el siguiente:

1. A genera un número aleatorio r, y envía a B $t = r^2$.
2. B elige un bit b al azar, y lo envía a A.
3. A calcula $u = r \cdot s^b$ y lo envía a A.

4. B comprueba que $u^2 = t \cdot ID^b$.

Al igual que antes, si A conoce s responderá satisfactoriamente el desafío de B.
Si A no conoce s, en el paso primero puede enviar a B, bien t, bien $t' = r^2 \cdot ID^{-1}$. Elija la opción que elija, tiene una probabilidad del 50 % de superar el desafío de B.

5 Protocolos de transferencia inconsciente o trascordada.

La idea básica de este tipo de protocolos es que A puede transmitir a B cierta información, o no. Pero A no sabe si la ha transmitido. Es decir, la probabilidad de que A transmita la información es p (pongamos $p = \frac{1}{2}$). La probabilidad entonces de que B la reciba es por tanto $\frac{1}{2}$. Obviamente, B sabe si la ha recibido o no, pero A no sabe con certeza si la ha enviado o no.
Una aplicación de esto puede ser si A es un administrador de varios secretos, y envía uno a B sin saber el que le ha enviado. Si A conociera el secreto que envía podría adquirir cierto poder sobre B. Vamos a continuación a describir dos protocolos de transferencia inconsciente.

5.1 *Protocolo de Rabin.*

La base de este protocolo es la misma que la del protocolo que describimos del lanzamiento de una moneda.
Supongamos que A tiene cierta información que va a transferir a B con probabilidad del 50 %. Entonces A envía a B esa información cifrada con un criptosistema RSA de llave pública (n, e), y envía a B la factorización de n mediante transferencia inconsciente. Para ello:

1. B elige un número al azar u entre 1 y $\frac{n-1}{2}$, calcula $v = u^2 (\text{mód } n)$ y se lo envía a A.
2. A calcula las cuatro raíces cuadradas de v módulo n, lo cual es posible pues A conoce la factorización de n. Elige una que sea menor que $\frac{n}{2}$ (hay dos en esa situación) y la envía a B.

Como ya vimos, B tiene un 50 % de probabilidad de factorizar n, en cuyo caso puede descifrar el mensaje. Por tanto, B recibe la información con un 50 % de probabilidad.
Por otra parte, A no tiene forma de saber (salvo que B se lo diga) si B ha conseguido factorizar n.

5.2 *Protocolo de Berger, Peralta y Tedrick.*

En este caso, el parámetro es un número n que es producto de dos primos grandes p y q tales que $p \equiv q \equiv 3 (\text{mód } 4)$. De esta forma nos garantizamos que cada par de raíces cuadradas gemelas de un resto cuadrático módulo n tienen símbolos de Jacobi opuestos.
Supongamos que A desea transferir de forma trascordada a B un mensaje m tal que $1 \leq m \leq \frac{n-1}{2}$. Entonces:

1. A envía a B n, y B comprueba que $\left(\frac{-1}{n}\right) = 1$.
2. B elige un número aleatorio x, entre 1 y n, calcula $w = x^2 (\text{mód } n)$ y se lo envía a A.
3. A calcula $b = \left(\frac{m}{n}\right)$ y $z = m^2 (\text{mód } n)$, y se los envía a B.
4. A calcula las raíces de $z \cdot w$ y envía una a B (llamémosla s).
5. B comprueba que $z \cdot w \equiv s^2 (\text{mód } n)$.

Una vez llegados aquí, B puede obtener m' como $m' = s \cdot x^{-1} (\text{mód } n)$ o $m' = -s \cdot x^{-1} (\text{mód } n)$ (elige de los dos el que sea menor que $\frac{n}{2}$).
Si $\left(\frac{m'}{n}\right) = b$ entonces m' es el mensaje. Si $\left(\frac{m'}{n}\right) = -b$ entonces no ha recibido el mensaje.

6 Protocolos de compromiso con un bit

Con este tipo de protocolos se pretende que un usuario A se comprometa con cierta información (por ejemplo un bit, aunque no es necesario) ante otro usuario B, o más usuarios, de forma que A no tiene que revelar esa información hasta que no haya transcurrido un cierto tiempo. Por otra parte, A no puede cambiar la información comprometida una vez que ha lanzado el compromiso.
Supongamos que s es la información que tiene A. Entonces A envía a B el compromiso $C(s)$, de forma que:

- B no puede recuperar el valor de s a partir del compromiso.
- A puede descubrir el compromiso revelando el valor de s a B. Con esto B puede comprobar la validez del mismo.
- A no puede cambiar el compromiso, es decir, no puede revelar un valor $s' \neq s$ tal que $C(s') = C(s)$.

Por ejemplo, se podría utilizar una función resumen como forma de calcular el compromiso. A podría comprometerse a que el bit i-ésimo de s es uno determinado.
Otro algoritmo, basado en el problema del logaritmo discreto es el que proponemos a continuación.

6.1 *Protocolo de Brassard, Crépeau y Chaum.*

En este protocolo, los parámetros son:

p un número primo grande.

q otro primo grande de forma que $p \equiv 1 (\text{mód } q)$.

α un elemento de orden q en $\mathbb{Z}_p$.

Si la información s es la que desea comprometer A, lo que hace es calcular $C(s) = \alpha^s (\text{mód } p)$, y enviar este valor a B.
B, con el compromiso recibido no puede calcular s, pues eso supondría resolver un problema de logaritmo discreto.
A no puede cambiar el valor del secreto, pues al ser s un elemento de orden q, s es el único entero entre 1 y $q-1$ que verifica que $\alpha^s \equiv C(s) (\text{mód } p)$.