UNIVERSIDAD DEL AZUAY

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

Monografía Previa la obtención
del título de Ingeniero de Sistemas

Tema: La Criptografía en la Seguridad de Sistemas Informáticos

Autores:
Cesar Andrade
David Cedillo

Cuenca, Septiembre del 2004

Responsabilidad

Los criterios vertidos en este trabajo de investigación, son de exclusiva responsabilidad de los autores.

Cesar Andrade

David Cedillo

1: Tema:

La Criptografía en la Seguridad de Sistemas Informáticos

2: Justificación.

En la actualidad dentro de los bienes más preciados y que de alguna manera indican el éxito para una empresa, organizaciones militares, políticas o de otra índole es la calidad de la información que genera y su gestión. Uno de los retos más fascinantes dentro de un sistema de intercambio de información es la protección de esta.

Además no hay que olvidar que vivimos en una era de informática donde la información que tengamos nos da el poder sobre los aspectos cotidianos en los que nos desenvolvemos, igualmente si estos datos caen en manos equivocadas nos dejan vulnerables con respecto a quien los posea por esto la importancia de mantener nuestra información segura.

Debido principalmente al uso de Internet, la protección de la información se ha transformado en una necesidad y con ello se hace más frecuente el uso de la Criptografía.

La criptografía es una disciplina que ha sido considerada desde tiempos muy antiguos como medio para preservar la confidencialidad de la información.

En la actualidad la utilización de la criptografía nos permite cifrar y/o proteger ya sea un archivo, mensaje, etc. Lo que nos va a garantizar aspectos muy importantes dentro de la seguridad informática como son la confidencialidad, la integridad, la disponibilidad y el no repudio tanto del emisor como del receptor.

3: Objetivos.

3.1 Objetivo General

Obtener un conocimiento claro y profundo acerca de conceptos fundamentales de Criptografía y sus aplicaciones.

3.2 Objetivos Específicos:

- Conocer los diversos tipos de cifrado de datos y su utilización en diferentes casos.
- Conocer los algoritmos y sistemas de privacidad, autenticación y protección más utilizados.

- Demostración de aplicaciones informáticas en las que se utilizan los sistemas de encriptación estudiados.

4: Contenido.

4.1 Introducción Histórica a la Criptografía
- 4.1.1 Definiciones
- 4.1.2 Confidencialidad, Integridad y No Repudio
- 4.1.3 Criptosistemas
- 4.1.4 Esquema de un Criptosistema
- 4.1.5 Clasificación de los Criptosistemas
  - 4.1.5.1 Criptosistemas Simétricos
  - 4.1.5.2 Criptosistemas Asimétricos
- 4.1.6 Recomendaciones de Bacon
- 4.1.7 Recomendaciones de Kerkchoffs

4.2 Criptografía Clásica
- 4.2.1 Clasificación Criptografía Clásica
- 4.2.2 Sistema Escítala
- 4.2.3 Sistema Cesar
  - 4.2.3.1 Ejemplo Sistema Cesar
- 4.2.4 Cifrador por Transformación Afín
  - 4.2.4.1 Ejemplo del Cifrador por Transformación Afín
- 4.2.5 El Cifrador de Alberti
  - 4.2.5.1 Ejemplo Cifrador de Alberti
- 4.2.6 El Cifrador de Vigenère
  - 4.2.6.1 Ejemplo Cifrador de Vigenère

4.3 Criptografía Moderna
- 4.3.1 Clasificación de la Criptografía Moderna
- 4.3.2 Cifrado en Flujo
- 4.3.3 Cifrado en Bloque

4.1 Introducción Histórica a la Criptografía

Desde la edad antigua era necesario ocultar la información en época de guerra o de enemigos del régimen; un ejemplo de esto fueron los egipcios que ocultaban sus mensajes en la cabeza de los esclavos. Otro ejemplo esta en los Griegos que ocultaban los mensajes, escribiendo el mensaje en una tira de tela envuelta en un cilindro y luego rellenaban el espacio que sobraba con letras al azar con lo cual el mensaje era legible solo si el telar era enrollado en un cilindro del mismo tamaño.

A comienzos de la segunda guerra mundial y hasta la actualidad, los mensajes cifrados han jugado un papel destacado en la historia, protegiendo secretos militares, diplomáticos, espías, y en nuestros días en las comunicaciones y datos que viajan por Internet.

El siglo XX ha revolucionado la criptografía. Retomando el concepto de las ruedas concéntricas de Alberti, a principios de la centuria se diseñaron teletipos equipados con una secuencia de rotores móviles. Estos giraban con cada tecla que se pulsaba. De esta forma, en lugar de la letra elegida, aparecía un signo escogido por la máquina según diferentes reglas en un código polialfabético complejo. Estos aparatos, se llamaron traductores mecánicos. Una de sus predecesoras fue la Rueda de Jefferson, el aparato mecánico criptográfico más antiguo que se conserva.

La primera patente data de 1919, y es obra del holandés Alexander Koch, que comparte honores con el alemán Arthur Scherbius, el inventor de Enigma una máquina criptográfica que los nazis creyeron inviolable, sin saber que a partir de 1942, propiciaría su derrota. En efecto, en el desenlace de la contienda, hubo un factor decisivo y apenas conocido: los aliados eran capaces de descifrar todos los mensajes secretos alemanes.

Una organización secreta, en la que participó Alan Turing, uno de los padres de la informática y de la inteligencia artificial, había logrado desenmascarar las claves de Enigma, desarrollando más de una docena de artilugios -las bombas- que desvelaban los mensajes cifrados. La máquina alemana se convertía así en el talón de Aquiles del régimen, un topo en el que confiaban y que en definitiva, trabajaba para el enemigo. La existencia de este conocimiento convirtió en uno de los secretos mejor guardados por los estadounidenses hasta varios años después de terminada la guerra.

4.1.1 Definiciones

La Real Academia Española define criptografía (oculto + escritura) como:

"el arte de escribir mensajes con una clave secreta o de modo enigmático".

Utiliza las matemáticas como principal herramienta y en la actualidad la informática y la telemática, que valiéndose de métodos y técnicas con el objeto principal de cifrar y/o proteger un mensaje o archivo por medio de un algoritmo complejo, usando una o más claves. Esto da lugar a diferentes tipos de sistemas de cifra que

permiten asegurar estos aspectos de la seguridad informática: la confidencialidad, la integridad, la disponibilidad y el no repudio tanto del emisor como del receptor.

4.1.2 Confidencialidad, Integridad y No Repudio

Estos son principios informáticos que se deben tomar en cuenta para tener una buena seguridad de los datos.

En el campo del Internet donde el intercambio de información es una de los objetivos principal de éste, y además no conocemos por donde viajarán los datos, es mediante la **confidencialidad** que aplicamos el ocultamiento selectivo de la información, para evitar la amenaza de revelación, esto implica:

- debe ser fácil su cifrado
- debe ser fácil su descifrado si se dispone de la clave
- difícil ("imposible") descifrar si no se dispone de la clave

La **integridad** implica establecer un mecanismo para verificar si un documento ha sido alterado, esto no quiere decir que evite la alteración pero si nos permite saber si esto sucedió en algún instante mientras viajaba el mensaje hacia el destinatario.

Mientras que el **No-Repudio** es un mecanismo por el cual un agente que efectúa una acción no puede posteriormente negar haberla efectuado, este es un requisito indispensable para posibilitar la digitalización de circuitos administrativos, utilización del documento digital, despapelización de las oficinas, transacciones a distancia. Para ello se requiere:

- Identificación autenticada del agente emisor.
- Verificación de integridad del documento (digital).

Cabe destacar que para la seguridad informática no se puede descuidar ninguno de las tres características, ya que si una faltase toda la seguridad estaría comprometida.

4.1.3 Criptosistemas

Un criptosistema, es un sistema que toma información entendible y mediante un proceso definido lo convierte en un mensaje completamente distinto y en teoría incomprensible para todos los que no conozcan como interpretarlo.

Un criptosistema es una quíntupla (M,C,K,E,D) que satisface las siguientes condiciones:

1) **M =** textos originales o base**.**

2) **C** = textos cifrados o codificados.

3) **K** = espacio de llaves, es decir, un conjunto finito de posibles llaves.

4) Para cada $\mathbf{k} \in \mathbf{K}$, existe una regla de cifrado $\mathbf{e_k} \in \mathbf{E}$ y su correspondiente regla de descifrado $\mathbf{d_k} \in \mathbf{D}$.

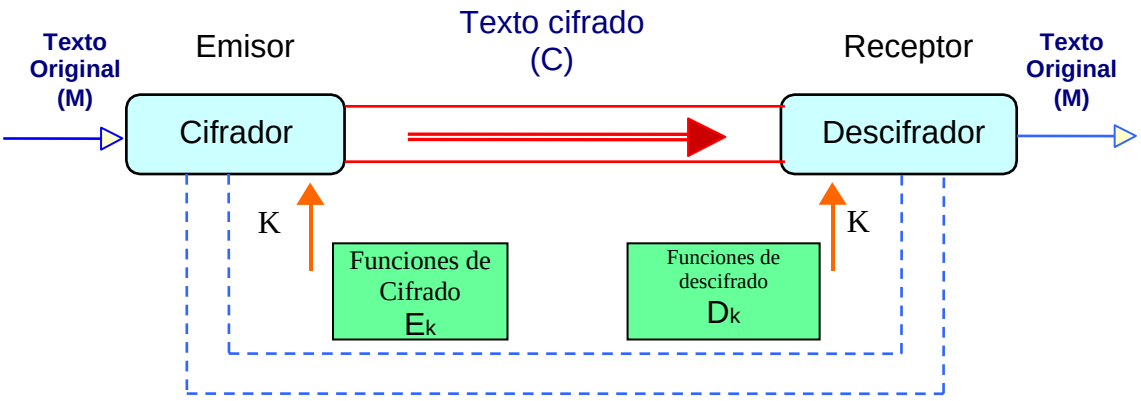
Si se tienen las funciones:

$$\mathbf{e_k : M \rightarrow C} \quad \mathbf{y} \quad \mathbf{d_k : C \rightarrow M}$$

Luego se cumple que: $d_k(e_k(M)) = M$ (para todo texto base).

La propiedad 4 es la principal, pues este nos indica que si el texto original **M** es encriptado usando la función $\mathbf{e_k}$, y el texto cifrado resultante es descifrado usando la función $\mathbf{d_k}$, luego el resultado será el texto original **M**.

4.1.4 Esquema de un Criptosistema



M= Textos Originales
C= Textos Cifrados
K= Claves
$E_K(M)$= Transformaciones de Cifrado
D_K (C)= Transformaciones de Descifrado

Figura 4.1.4 Esquema de Criptosistema

4.1.5 Clasificación de los Criptosistemas

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

4.1.5.1 Criptosistemas simétricos.- También denominados criptosistemas de Clave Privada, en el cual existirá una única clave (secreta) que deben compartir emisor y receptor. Con la misma clave se cifra y se descifra por lo que la seguridad se encuentra sólo en mantener dicha clave en secreto.

4.1.5.2 Criptosistemas asimétricos.- También llamados Criptosistemas de Clave Publica, en estos criptosistemas cada usuario crea un par de claves, una privada y otra pública, inversas dentro de un cuerpo finito. Lo que se cifra en emisión con una clave, se descifra en recepción con la clave inversa. La seguridad del sistema reside en la dificultad computacional de descubrir la clave privada a partir de la pública. Para ello, usan funciones matemáticas de un solo sentido con trampa.

4.1.6 Recomendaciones de Bacon

Filósofo y estadista inglés del siglo XVI:

- Dado un texto en claro M y un algoritmo de cifra Ek, el cálculo de Ek(M) y su inversa debe ser sencillo.
- Será imposible encontrar el texto en claro M a partir del criptograma C si se desconoce la función de descifrado Dk.
- El criptograma deberá contener caracteres distribuidos para que su apariencia sea inocente y no dé pistas a un intruso.

4.1.7 Recomendaciones de Kerckhoffs

Profesor holandés en París del siglo XIX

- El sistema debe ser en la práctica imposible de criptoanalizar.
- Las limitaciones del sistema no deben plantear dificultades a sus usuarios.
- Método de elección de claves fácil de recordar.
- El criptógrafo debe ser portable.
- No debe existir una larga lista de reglas de uso.

4.2 Criptografía Clásica

La criptografía clásica abarca desde tiempos inmemoriales, como veremos a continuación, hasta la mitad del siglo XX. El término de clásica se debe tanto a las técnicas utilizadas, básicamente operaciones de sustitución y transposición de caracteres, con o sin clave pero siempre unido al concepto de clave secreta, como al uso de máquinas dedicadas a la cifra. En el caso de los sistemas modernos, éstos hacen uso, además de lo anterior, de algunas propiedades matemáticas como, por ejemplo, la dificultad del cálculo del logaritmo discreto o el problema de la factorización de grandes números. No obstante, muchos sistemas modernos y que en la actualidad se siguen utilizando, como los algoritmos de clave secreta DES e IDEA, se basan en conceptos que podríamos denominar clásicos como son los de transposición y sustitución con una clave privada, si bien en estos sistemas la operación se realiza sobre una cadena de bits y no sobre caracteres.

Muchos de los criptosistemas clásicos, en particular aquellos que transforman el mensaje en claro aplicando técnicas de sustitución y transposición, basan su seguridad principalmente en el secreto de la transformación o algoritmo de cifra. Es ésta también una diferencia fundamental con respecto a los sistemas modernos, en los que el algoritmo se hace público puesto que la fortaleza del sistema reside en la imposibilidad computacional de romper una clave secreta. Observe que el hacer público el algoritmo de cifra permite al criptólogo evaluar la calidad del software desarrollado, en tanto será estudiado por la comunidad científica intentando buscar un defecto, una puerta falsa, una rutina innecesaria, una codificación no depurada, etc.

4.2.1 Clasificación Criptografía Clásica

Como explicamos anteriormente los métodos clásicos son aquellos en los que, además de las máquinas dedicadas para cifrar, se usan por separado técnicas de sustitución y transposición aplicadas a los caracteres del texto en claro. Las técnicas criptográficas utilizadas en este caso son en su totalidad orientadas a sistemas de clave secreta, generalmente manteniendo también en secreto el algoritmo, incluso en el caso en que el cifrador cuente con una clave secreta. La operación de cifra se realiza sobre caracteres alfanuméricos, por lo general alfabéticos, y en ese mismo formato se transmiten o almacenan.

Existen dos técnicas básicas orientadas a caracteres:

- Transposición: los caracteres o letras del mensaje se redistribuyen sin modificarlos y según unas reglas, dentro del criptograma. También se le conoce como permutación.
- Sustitución: un caracter o letra se modifica o sustituye por otro elemento en la cifra.

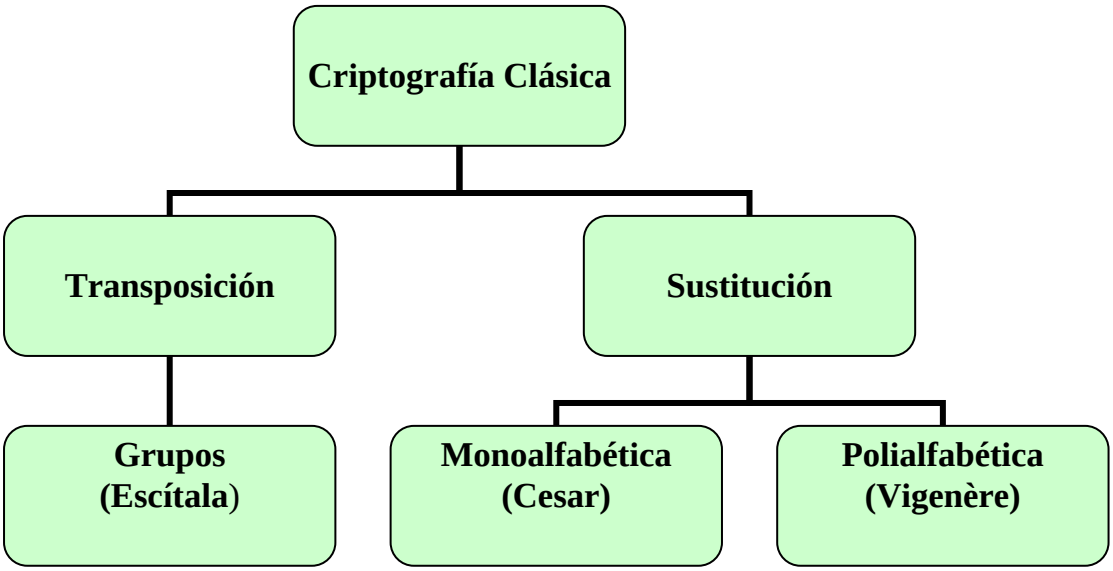


Figura 4.2.1 Clasificación de la Criptografía Clásica

Los cifradores por **transposición** utilizan la técnica de permutación de forma que los caracteres del texto se reordenan mediante un algoritmo específico. Un caso representativo de esta transformación sería transmitir el mensaje en bloques de cinco caracteres pero reordenados de forma que su posición en el criptograma sea, por ejemplo, 43521; es decir, el cuarto caracter del bloque en claro se transmite primero, a continuación el tercero, después el quinto, luego el segundo y, por último, el primero. Esta operación se repetirá en cada bloque de 5 caracteres del mensaje. Por lo tanto, la transposición implica que los caracteres del criptograma serán exactamente los mismos que los del texto en claro.

Ejemplo: Utilizando permutación 43521 cifrar el siguiente mensaje, tomando bloques de 5 caracteres.

M = CRIPTOGRAFIA EN LA UNIVERSIDA DEL AZUAY

Solución:

- Separamos en bloques de 5 caracteres

M = CRIPT OGRAF IAENL AUNIV ERSID ADELA ZUAYX

- Aplicamos la permutación 43521 para cifrar M

M = CRIPT OGRAF IAENL AUNIV ERSID ADELA ZUAYX

C = PITRC ARFGO NELAI INVUA ISDRE LEADA YAXUZ

Los cifradores por **sustitución** utilizan la técnica de modificación de cada caracter del texto en claro por otro correspondiente al alfabeto de cifrado. Si el alfabeto de cifrado es el mismo que el del mensaje o bien único, hablamos entonces de cifradores **monoalfabéticos**; en el cifrador del César, por ejemplo, la letra *A* del texto en claro se cifraba siempre como la letra *D*, es decir, existe un único alfabeto en la operación de transformación del mensaje en criptograma.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
M =	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

Cesar

A =	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

Ejemplo:

M= CIFRADO CESAR

Solución:

C= FLIUDGR FHVDU

Por el contrario, si en dicha operación intervienen más de un alfabeto, se dice que el cifrador es **polialfabético**.Una forma de utilizar varios alfabetos es la siguiente, si tenemos un mensaje los caracteres que se encuentren en las posiciones impares podemos aplicar un desplazamiento de 15 espacios a la derecha del alfabeto, y a los caracteres pares podemos aplicar un desplazamiento de 10 espacios también hacia la derecha del alfabeto. De esta forma una letra A que se encuentre en una posición impar va a tener un valor diferente de otra A que se encuentre en una posición par.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**
$M_{i=}$	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

Impares

$A_{1=}$	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

Pares

$A_{2=}$	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

Ejemplo:

M= CIFRADO POLIALFABETICO

Solución:

C= QRTBONY EYZROUTOLSDWMD

La principal amenaza criptoanalítica proviene de la alta redundancia de la fuente. Shannon sugirió por ello dos métodos básicos para frustrar un criptoanálisis estadístico: la difusión y la confusión.

El propósito de la **difusión** consiste en anular la influencia de la redundancia de la fuente sobre el texto cifrado. Hay dos formas de conseguirlo. La primera, conocida como transposición, evita los criptoanálisis basados en las frecuencias de las n-palabras. La otra manera consiste en hacer que cada letra del texto cifrado dependa de un gran número de letras del texto original.

El objetivo de la **confusión** consiste en hacer que la relación entre la clave y el texto cifrado sea lo más compleja posible, haciendo así que las estadísticas del texto cifrado no estén muy influidas por las del texto original. Eso se consigue normalmente con la técnica de la sustitución. En solitario, ni confusión ni difusión constituyen buenas técnicas de cifrado.

4.2.2 Sistema Escítala

La escítala era usada en el siglo V a.C. por el pueblo griego de los lacedemonios. Consistía en un bastón en el que se enrollaba una cinta de cuero y luego se escribía en ella el mensaje de forma longitudinal y se rellenaba el espacio vacío con letras al azar. Al desenrollar la cinta, las letras aparecen desordenadas. *(Ver figura 4.2.2)*

La única posibilidad de recuperar el texto en claro pasaba por enrollar dicha cinta en un bastón con el mismo diámetro que el usado en el extremo emisor y leer el mensaje de forma longitudinal. La clave del sistema está en el diámetro del bastón. Se trata de una cifra por transposición pues los caracteres del criptograma son los mismos que en el texto en claro distribuidos de otra forma.

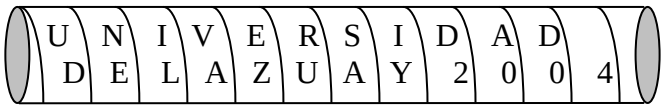


Figura 4.2.2 La Escítala

Texto cifrado (C): UD NE IL VA EZ R USA IY D2 A0 D0 4

Texto en claro (M): UNIVERSIDAD DEL AZUAY 2004

4.2.3 Sistema Cesar

En el siglo I a.C., Julio César usa este cifrador monoalfabético, cuyo algoritmo consiste en el desplazamiento de tres espacios hacia la derecha de los caracteres del texto en claro (*Ver figura 4.2.3*). Es un cifrador por sustitución monoalfabético en el que las operaciones se realizan módulo **N**, siendo **N** el número de elementos del alfabeto (en aquel entonces latín).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Mi	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
Ci	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

Figura 4.2.3 Alfabeto de cifrado del César para castellano mod 27

Ci = Mi + 3 mod 27

Cada letra se cifrará siempre igual. Es una gran debilidad y hace que este sistema sea muy vulnerable y fácil de atacar simplemente usando las estadísticas del lenguaje.

4.2.3.1 Ejemplo Sistema Cesar

A continuación vamos a cifrar un mensaje mediante el Sistema Cesar. El mensaje es el siguiente: “ESCUELA DE INGENIERIA DE SISTEMAS ”

M = ***ESCUELA DE INGENIERIA DE SISTEMAS***

Solución:

Ci = Mi + 3 mod 27

Luego sustituimos las letras del alfabeto por su correspondiente y obtenemos un texto cifrado:

C = ***HVFXHÑD GH LPJHPLHULD GH VLVWHODV***

4.2.4 Cifrador por Transformación Afín

Este es un cifrador monoalfabético y decimos que un cifrador es por transformación afín si se cumple que la constante de decimación *a* es mayor que 1 y la constante de desplazamiento *b* distinto de cero.

$$\mathbf{C_i = (a*M_i + b) \bmod N}$$

$$\mathbf{M_i = a^{-1} (C_i - b) \bmod N}$$

Siendo **a** y **b** dos números enteros menores que el cardinal **N** del alfabeto, y cumpliendo que **mcd (a, N) = 1**

La clave de cifrado **K,** viene entonces dada por el par (a, b). El algoritmo de César será pues una transformación afín con una clave **K** = (1, 3).

4.2.4.1Ejemplo del Cifrador por Transformación Afín

En este ejemplo vamos a hallar el alfabeto de cifrado utilizando los siguientes datos:

K = (7,3)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Mi	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

Solución: $C_i = (7*M_i + 3) \bmod 27$

$C_0 = (7*A + 3) \bmod 27 = (7*0 + 3) \bmod 27 = 3 =$ D
$C_1 = (7*B + 3) \bmod 27 = (7*1 + 3) \bmod 27 = 10 =$ K
$C_2 = (7*C + 3) \bmod 27 = (7*2 + 3) \bmod 27 = 17 =$ Q
$C_3 = (7*D + 3) \bmod 27 = (7*3 + 3) \bmod 27 = 24 =$ X
$C_4 = (7*D + 3) \bmod 27 = (7*4 + 3) \bmod 27 = 4 =$ E

. .
. .
. .

$C_{25} = (7*Y + 3) \bmod 27 = (7*25 + 3) \bmod 27 = 16 =$ P
$C_{26} = (7*Z + 3) \bmod 27 = (7*26 + 3) \bmod 27 = 23 =$ W

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Mi	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
Ci	D	K	Q	X	E	L	R	Y	F	M	S	Z	G	N	T	A	H	Ñ	U	B	I	O	V	C	J	P	W

4.2.5 El Cifrador de Alberti

En el siglo XVI León Battista Alberti presenta un manuscrito en el que describe un disco cifrador con el que es posible cifrar textos sin que exista una correspondencia única entre el alfabeto del mensaje y el alfabeto de cifrado como en los casos analizados anteriormente. Con este sistema, cada letra del texto en claro podía ser cifrada con un caracter distinto dependiendo esto de una clave secreta. Se dice entonces que tales cifradores usan más de un alfabeto por lo que se denominan **cifradores polialfabéticos**, a diferencia de los anteriores denominados **monoalfabéticos**.

Como se aprecia en la *Figura 4.2.4*, el **disco de Alberti** presenta en su círculo exterior los 20 caracteres del latín, esto es, los mismos del alfabeto castellano excepto las letras H, J, Ñ, K, U, W e Y, y se incluyen los números 1, 2, 3 y 4 para códigos especiales. Por su parte, en el disco interior aparecen todos los caracteres del latín además del signo & y las letras H, K e Y. Al ser 24 los caracteres representados en cada disco, es posible definir hasta 24 sustituciones diferentes; es decir, dependiendo de la posición del disco interior la cantidad máxima de alfabetos de cifrado es igual a 24. Luego, para cifrar un mensaje, una vez establecida la correspondencia entre caracteres de ambos discos o, lo que es lo mismo, el alfabeto de cifrado, se repasa letra a letra el texto en claro del disco exterior y se sustituye cada una de ellas por la letra correspondiente del disco interior.

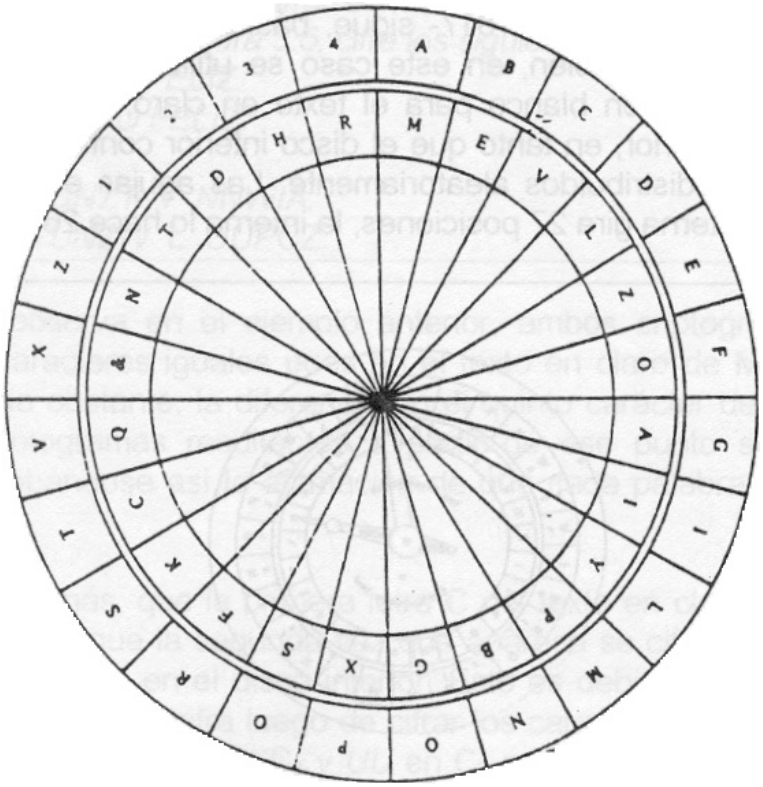


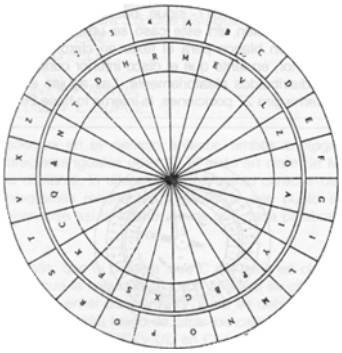
Figura 4.2.5 Disco cifrador de Alberti.

La innovación que supone este sistema consiste en que el alfabeto de sustitución puede ser cambiado durante el proceso de cifrado, por ejemplo cada k caracteres, simplemente girando el disco interior y por tanto utilizando otro alfabeto de sustitución.

4.2.5.1 Ejemplo Cifrador de Alberti

En el siguiente ejemplo vamos a cifrar un mensaje utilizando el disco cifrador de Alberti, para esto vamos a hacer coincidir el número 1 del disco exterior con el signo & del disco interior.

M = DISCO CIFRADOR DE ALBERTI



Solución:

Desplazamos el disco interior dos espacios en el sentido de las agujas del reloj y leemos el caracter cifrado en el disco interior bajo el caracter correspondiente del texto en claro del disco exterior, obteniéndose:

C=EOSMP MOLXHEPX EV HARVXFO

4.2.6 El Cifrador de Vigenère

El cifrador polialfabético más conocido es el sistema de Vigenère, así denominado en honor al criptólogo francés **Blaise de Vigenère** (1523-1596). Este tipo de criptosistemas aparecieron para sustituir a los monoalfabéticos o de sustitución simple, basados en el Cesar, que presentaban ciertas debilidades frente al ataque de los criptoanalistas relativas a la frecuencia de aparición de elementos del alfabeto.

Cifrado de Vigenère para castellano mod 27

$$Ci = Mi + Ki \bmod 27$$

El principal elemento de este sistema es la llamada Tabla de Vigenère, una matriz de caracteres cuadrada.

	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
A	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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

Figura 4.2.6 Tabla de Vigenère

Usa una clave **K** de longitud L y cifra caracter a caracter sumando módulo **N** el texto en claro con los elementos de esta clave.

La clave del sistema de cifrado de Vigenère es una palabra de letras, **K**, del alfabeto utilizado anteriormente.

De esta forma, el mensaje a cifrar en texto claro ha de descomponerse en bloques de elementos letras y aplicar sucesivamente la clave empleada a cada uno de estos bloques, utilizando la tabla anteriormente proporcionada.

Para realizar el descifrado se procede en forma inversa. Es decir escogemos el carácter Ki de la clave y lo ubicamos en la fila de la tabla, buscamos por esa fila el otro caracter del texto cifrado Ci, una vez hallado subimos por esa columna hasta llegar a la letra de la primera fila, este será el caracter descifrado Mi.

4.2.6.1 Ejemplo del Cifrador de Vigenère

A continuación vamos a cifrar el siguiente mensaje “ESCUELA DE INGENIERIA DE SISTEMAS”, en el cual utilizaremos como palabra clave “UDA”:

M = “ESCUELA DE INGENIERIA DE SISTEMAS”
K = “UDA”

Ci = Mi + Ki mod 27

Solución:

En primer lugar, nos fijamos en la longitud de la clave: es de tres caracteres, por lo que descomponemos la frase en bloques de longitud 3; aunque el último bloque es de longitud dos, esto no afecta para nada al proceso de cifrado:

M=	ESC	UEL	ADE	ING	ENI	ERI	ADE	SIS	TEM	AS

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

Mi	ESC	UEL	ADE	ING	ENI	ERI	ADE	SIS	TEM	AS
Ki	UDA	UDA	UDA	UDA	UDA	UDA	UDA	UDA	UDA	UDA
Ci	YVA	OHL	UGE	CPG	YPI	YUI	UGE	JLS	ÑHM	UV

Por ejemplo, la primera letra del texto cifrado “Y” = **Ci** corresponde a la entrada, (Mi=E; Ki=U) equivalentemente, de la tabla de Vigenère.

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

Para la operación de descifrado utilizamos la siguiente formula:

Mi = (Ci - Ki) mod N

4.3 Criptografía Moderna

Los criptosistemas modernos, cuya cifra en bits está orientada a todos los caracteres ASCII o ANSI usan por lo general una operación algebraica en Zn, un cuerpo finito, sin que necesariamente este módulo deba corresponder con el número de elementos del alfabeto o código utilizado. Es más, nunca coinciden; siempre será mucho mayor el cuerpo de trabajo que el alfabeto.

Su fortaleza está en la imposibilidad computacional de descubrir una clave secreta única, en tanto que el algoritmo de cifra es o debería ser público.

4.3.1 Clasificación de la Criptografía Moderna

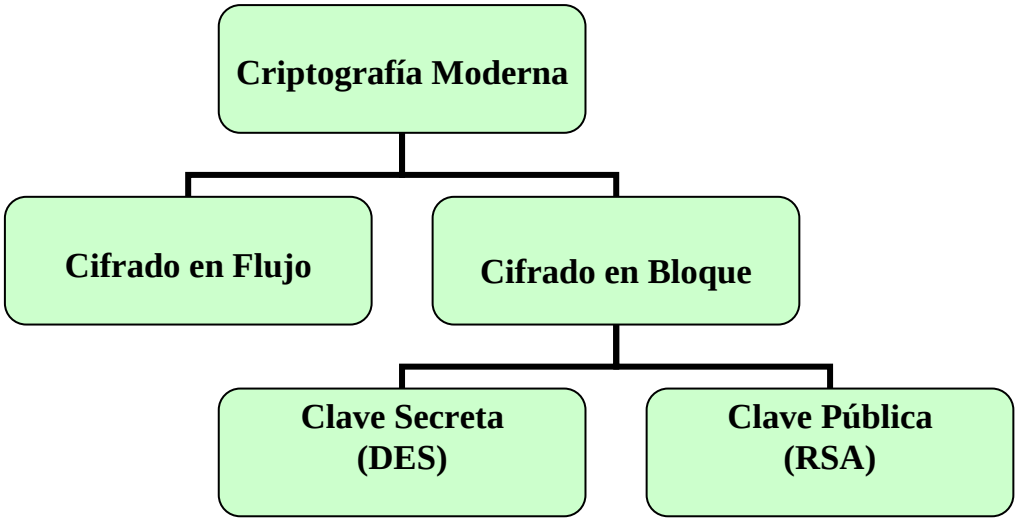


Figura 4.3.1 Clasificación Criptografía Moderna

4.3.2 Cifrado en Flujo

En 1917, J. Mauborgne y G. Vernam inventaron un criptosistema perfecto según el criterio de Shannon. Dicho sistema consistía en emplear una secuencia aleatoria de igual longitud que el mensaje, que se usara una única vez, lo que se conoce en inglés como one-time pad, combinándola mediante alguna función simple y reversible usualmente el OR exclusivo (XOR), con el texto en claro caracter a caracter. Este método presenta el grave inconveniente de que la clave es tan larga como el propio mensaje, y si disponemos de un canal seguro para enviar la clave, ¿por qué no emplearlo para transmitir el mensaje directamente? Evidentemente, un sistema de Vernam carece de utilidad práctica en la mayoría de los casos, pero supongamos que disponemos de un generador pseudoaleatorio capaz de generar secuencias criptográficamente aleatorias, de forma que la longitud de los posibles ciclos sea extremadamente grande. En tal caso podríamos, empleando la semilla

del generador como clave, obtener cadenas de bits de usar y tirar, y emplearlas para cifrar mensajes simplemente aplicando la función XOR entre el texto en claro y la secuencia generada. Todo aquel que conozca la semilla podría reconstruir la secuencia pseudo aleatoria y de esta forma descifrar el mensaje. *(Ver fig 4.3.2)*

Uno de los primeros criptosistemas que explotaban la idea del generador pseudoaleatorio fue el cifrado de Lorenz, empleado por Alemania en la II Guerra Mundial, junto con la máquina ENIGMA. Este sistema se basaba en un dispositivo que generaba una secuencia supuestamente imposible de reproducir sin conocimiento de la clave, que se combinaba con los mensajes para obtener los criptogramas. Por suerte para los expertos de Bletchley Park, estas secuencias presentaban una sutil estructura que podría ser analizada mediante técnicas estadísticas. Para ello un equipo de científicos, entre los que se encontraba Max Newman puso en práctica las ideas de Alan Turing y desarrolló Colossus, el primer computador de la Historia, capaz de descifrar los mensajes codificados mediante el sistema Lorenz.

Los algoritmos de cifrado en flujo no son más que la especificación de un generador pseudoaleatorio, y permiten cifrar mensajes de longitud arbitraria, combinando el mensaje con la secuencia mediante la operación OR exclusivo byte a byte, en lugar de dividirlos en bloques para codificarlos por separado. Como sería de esperar, estos criptosistemas no proporcionan seguridad perfecta, ya que mientras en el cifrado de Vernam el número de posibles claves era tan grande como el de posibles mensajes, cuando empleamos un generador tenemos como mucho tantas secuencias distintas como posibles valores iniciales de la semilla.

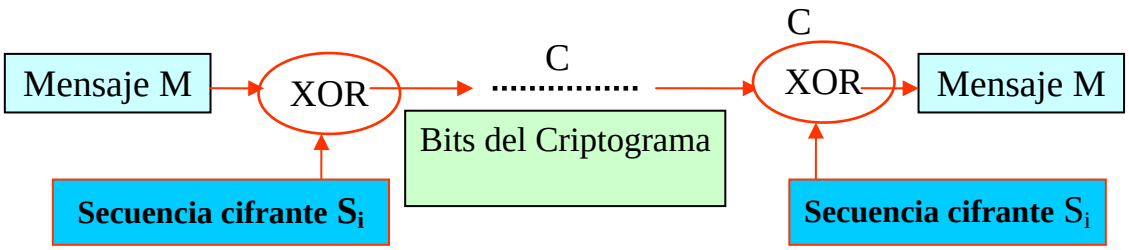


Figura 4.3.2 Cifrado en Flujo

4.3.3 Cifrado en Bloque

La gran mayoría de los algoritmos de cifrado simétricos se apoyan en los conceptos de confusión y difusión inicialmente propuestos por Shannon, que se combinan para dar lugar a los denominados cifrados de producto, en el cual el mensaje en claro se divide en bloques y se aplica el algoritmo sobre cada uno de forma independiente con la misma clave de manera que el cifrado de cada símbolo depende de los demás símbolos del mismo bloque, para descifrar una parte no es preciso descifrar el mensaje completo. *(Ver fig. 4.3.3)*

Recordemos que la confusión consiste en tratar de ocultar la relación que existe entre el texto claro, el texto cifrado y la clave. Un buen mecanismo de confusión haría demasiado complicado extraer relaciones estadísticas entre las tres cosas. Por su parte la difusión trata de repartir la influencia de cada bit del mensaje original lo más posible entre el mensaje cifrado. Hemos de hacer notar que la confusión por sí sola sería suficiente, ya que si establecemos una tabla de sustitución completamente diferente para cada clave con todos los textos claros posibles tendremos un sistema extremadamente seguro. Sin embargo, dichas tablas ocuparían cantidades astronómicas de memoria, por lo que en la práctica serían inviables. Por ejemplo, un algoritmo que codificara bloques de 128 bits empleando una clave de 80 bits necesitaría una tabla de aproximadamente 1063 entradas. Lo que en realidad se hace para conseguir algoritmos fuertes sin necesidad de almacenar tablas enormes es intercalar la confusión (sustituciones simples, con tablas pequeñas) y la difusión (permutaciones). Esta combinación se conoce como cifrado de producto. La mayoría de los algoritmos se basan en diferentes capas de sustituciones y permutaciones, estructura que denominaremos Red de sustitución-permutación. En muchos casos el criptosistema no es más que una operación combinada de sustituciones y permutaciones, repetida n veces, como ocurre con DES.

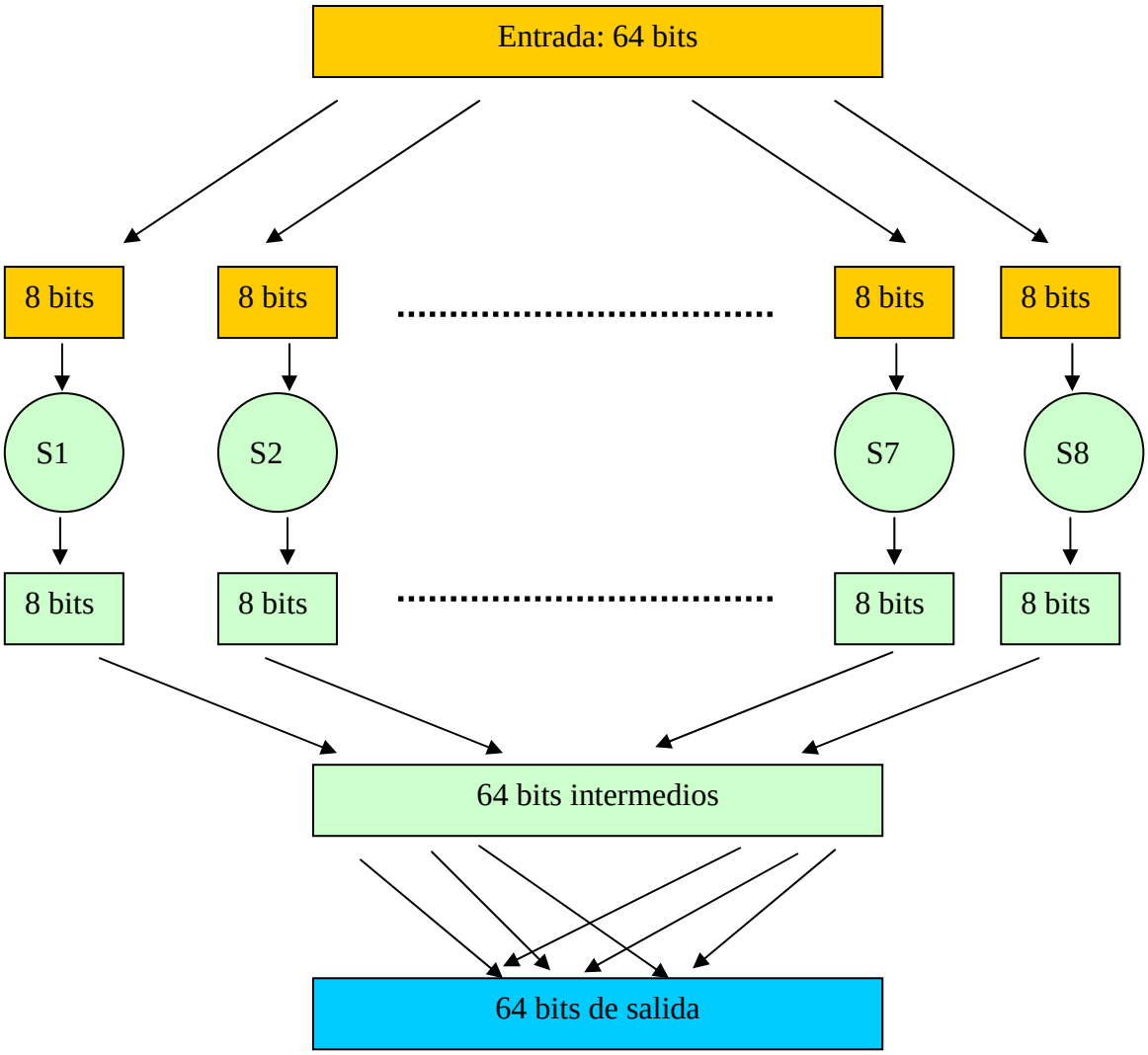


Figura 4.3.3 Cifrado en Bloque

4.4 Criptografía Clave Secreta o Simétrica

En los cifrados de clave secreta, la seguridad depende de un secreto compartido exclusivamente por emisor y receptor, si un atacante descubre la clave utilizada en la comunicación, ha roto el criptosistema.

Hasta la década de los setenta, la invulnerabilidad de todos los sistemas dependía de este mantenimiento en secreto de la clave de cifrado. Este hecho presentaba una gran desventaja: había que enviar, aparte del criptograma, la clave de cifrado del emisor al receptor, para que éste fuera capaz de descifrar el mensaje. Por tanto, se incurría en los mismos peligros al enviar la clave, por un sistema que había de ser supuestamente seguro, que al enviar el texto plano.

Dado el hecho de que exista al menos una clave de cifrado/descifrado entre cada dos usuarios de un sistema haría inviable la existencia de criptosistemas simétricos en las grandes redes de computadores de hoy en día: para un sistema de computación con usuarios, se precisarían claves diferentes, lo cual es obviamente imposible en grandes sistemas. Todos estos motivos han propiciado que el estudio de los cifradores simétricos (excepto DES) quede relegado a un papel histórico.

Por otro lado la velocidad de cifra es muy alta y por ello se usará para realizar la función de cifra de la información. Además, con claves de sólo unas centenas de bits obtendremos una alta seguridad pues su no linealidad y algoritmo hace que el único ataque que puede prosperar sea el de la fuerza bruta.

A finales de los años cuarenta, Shannnon sugirió nuevas ideas para futuros sistemas de cifrado. Sus sugerencias se referían al uso de operaciones múltiples que mezclaran transposiciones y sustituciones.

4.4.1 Redes de Feistel

Horst Feistel: inventor (IBM) del algoritmo LUCIFER a comienzos de los años 70. El algoritmo fue utilizado por el Reino Unido. En 1974 se propone a la NSA como estándar y en ese año dará origen al DES.

En criptografía existe un cierto tipo de estructura de cifrado llamado Redes de Feistel, y es utilizada en algunos algoritmos como DES, Lucifer, FEAL, CAST, Blowfish, etc.

Estas redes dividen el texto a cifrar, M, en dos mitades, L y R, y van cifrando iterativamente (repetitivamente :) una de las mitades, luego las mitades se intercambian y el proceso se vuelve a repetir. *(Ver figura 4.3.1)*

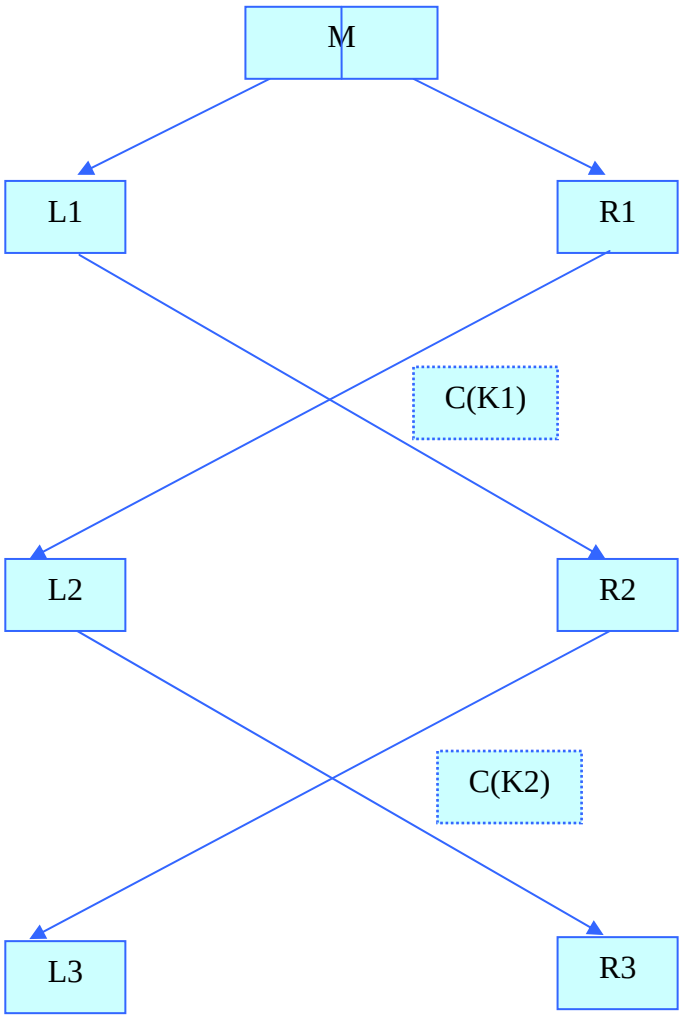


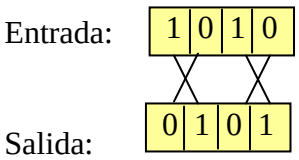
Figura 4.4.1 Red de Feistel

Como se puede observar, se usan varias claves C(K1,K2….), una por cada ciclo. Pues bien, esta red tiene la curiosa propiedad de ser reversible, si las K, se aplican en sentido inverso. En el DES se usan 16 Ks, por lo tanto el DES es un red de Feistel con 16 ciclos, pero tiene varias permutaciones que ahora veremos.

4.4.2 Permutaciones

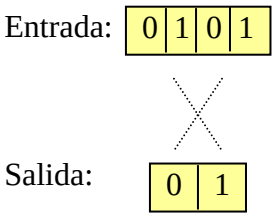
Una permutación es básicamente un reordenamiento de los bits del número a permutar. Para permutar un número se siguen unas tablas que dicen como se debe permutar. Hay tres "tipos" de permutaciones.

Las permutaciones en las que el número de la salida tiene el mismo número de bits que la entrada:



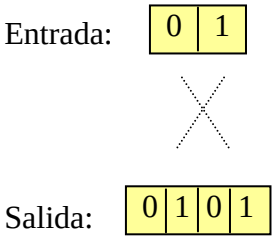
Los bits simplemente se cambian de sitio.

Otro tipo son las de compresión:



Aquí la salida ocupa menos bits que la entrada, con lo que algunos bits se pierden.

Y por último las de expansión:



Igual que las de compresión, pero al revés. Los bits se repiten para crear un número de bits mayor.

Luego veremos más detenidamente las permutaciones en el DES.

4.4.3 Las S-Boxes

S-BOXES quiere decir cajas de sustitución. En las permutaciones hemos visto como los bits se reordenaban, pero aquí los bits de la salida no tienen por que ser los mismos que la entrada. Las S-BOXES son unas "cajas" que toman una entrada de 6 bits y devuelven 4 (en el caso del DES).

Más concretamente son arrays bidimensionales, con 4 filas y 16 columnas.

Para obtener el índice en la fila se concatenan los bits 1 y 6, y para el índice en la columna se concatenan los bits 2, 3, 4 y 5.

Así por ejemplo:

Entrada: **1 0 0 0 1 0**

índice en la fila: **1 0** [2]

índice en la columna: **0 0 0 1** [1]

El grupo de cuatro bits que se encuentre en la posición [2][1] será la salida para la entrada: 1 0 0 0 1 0.

4.5 Criptosistema DES

Sin duda el cifrado en bloque más conocido es el llamado DES. Este sistema se puede catalogar como un cifrado en bloque que es a la vez un cifrado producto de transposiciones y sustituciones.

Su estructura general era del tipo Feistel, también usada en Lucifer. El algoritmo cifraba en bloques de 64 bits y utilizaba 16 iteraciones de un cifrado simple. La primera fortaleza del algoritmo venía dado por las denominadas Cajas-S; se trata de una operación de búsqueda sobre una tabla no lineal por la que grupos de 6 bits son reemplazados por grupos de 4 bits. Estas búsquedas en tablas se expresaban como cadenas de constantes.

NBS (National Bureau of Standards) no disponía de capacidad técnica para evaluar el algoritmo, por lo que solicitaron la ayuda de la **Agencia Nacional de Seguridad (NSA).** La NSA hizo dos cosas:

1.- Cambiaron las constantes de las Cajas-S

2.- Redujeron drásticamente el tamaño de la clave desde el original de 128 bits hasta 56 bits.

El algoritmo revisado fue llamado DES y publicado por el NBS en Marzo de 1975. Hubo considerables protestas públicas (los cambios que realizó la NSA no se hicieron públicos y no se dio ninguna explicación respecto de las constantes de las Cajas-S así como del motivo que llevo a reducir el tamaño de la clave). Originalmente se suponía que la longitud de la clave se reducía a 64 bits, pero cuando se publicó el estándar, convertía 8 de estos bits en "bits de paridad" usados para confirmar la integridad de los otros 56 bits, y no como parte de la clave.

A pesar de la críticas DES fue adoptado como un Estándar Federal de Procesamiento de Información en Noviembre de 1976. Era la primera vez que un algoritmo de cifrado evaluado por la NSA se hacía público.

Después de convertirse en un estándar del gobierno de los EE.UU., DES fue adoptado por otros organismo de estándares, incluyendo ANSI e ISO. Se convirtió en el algoritmo de cifrado estándar en la industria bancaria, y fue usado en muchas ampliaciones en todo el mundo. Los términos del estándar estipulaban que debía ser revisado y recertificado cada cinco años. NBS recertificó DES por primera vez en 1987. **NIST** (**National Institute for Standards and Technology),** como es conocido NBS tras cambiar su nombre, recertificó DES en 1993. En 1997 inició un programa para reemplazar a DES: el Estándar Avanzado de Cifrado (AES).

A finales de los 90, se extendió ampliamente la sospecha que la NSA era capaz de romper DES probando todas las claves posibles (llamado ataque por "fuerza bruta"). Esta posibilidad fue gráficamente demostrada por la Fundación de Fronteras Electrónicas (EFF) en Julio de 1998, cuando John Gilmore construyó una máquina por 250.000 dólares capaz de romper DES mediante fuerza bruta en sólo unos días.

4.5.1 Características del Criptosistema DES

DES es el algoritmo de cifrado más estudiado que se ha inventado jamás. Casi todos los nuevos algoritmos usados hoy en día se basan en los principio del DES y todavía en la actualidad se publican documentos analizando diferentes aspectos del DES.

Sus características son:

- Cifrador de bloque.
- Tipo Feistel
- Longitud de clave de 64 bits que se trunca a 56
- Realiza 16 vueltas.
- La cifra del bloque central usa técnicas de sustituciones y permutaciones.
- Para poder realizar las sumas OR exclusivo (XOR), usará permutaciones con expansión y compresión para igualar el número de bits
- En el descifrado se aplican claves y desplazamientos en sentido inverso.

En la siguiente gráfico *(Ver fig. 4.3.3)* se puede observar una visión muy general del algoritmo DES.

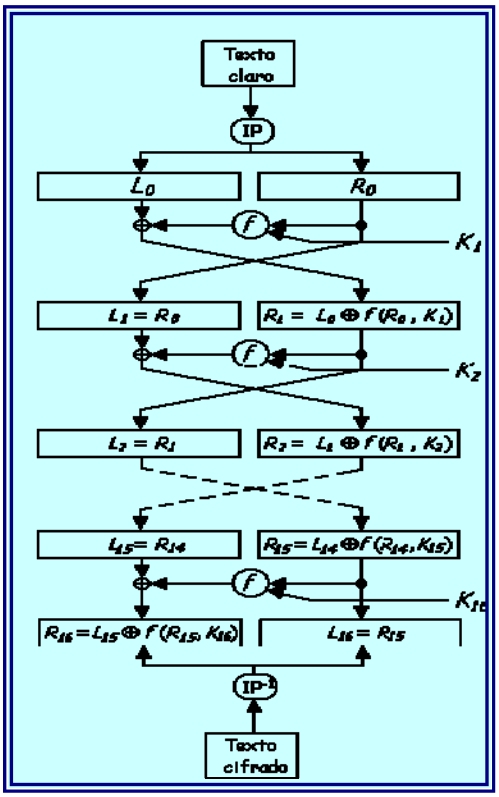


Figura 4.5.1 Algoritmo DES

4.5.2 Ejemplo del Criptosistema DES

Lo primero es poner la clave **K** a usar en el algoritmo. Se usa una clave de 64 bits de longitud. A esta clave se le aplica una permutación que la trunca a 56 bits, eliminando los bits de paridad (8, 16, 24, 32, 40, 48, 56, 64), y reordenando los restantes.

Para poder hacer esto utilizamos la siguiente matriz de permutación

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

Figura 4.5.2.1 Matriz de permutación PC-1

Ahora la clave de 56 bits se parte en dos de 28, y se van rotando cada parte independientemente de la otra dos bits hacia la izquierda en cada ciclo, menos para los ciclos 1, 2, 9 y último, que se desplaza solo uno, los desplazamientos los podemos observar en la *figura 4.5.2.2*.

Desplazamientos			
Vuelta i	Bits Desp. Izda.	Vuelta i	Bits Desp. Izda.
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

Figura 4.5.2.2 Tabla de desplazamientos

Ahora tenemos 16 claves de 56 bits. Cada una de estas claves se permuta siguiendo una permutación de compresión PC-2 *(Ver fig. 4.5.2.3)*, que la trunca a 48 bits, eliminando los bits 9, 18, 22, 25, 35, 38, 43, 54.

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

Figura 4.5.2.3 Matriz de Permutación de Compresión de 48 bits PC-2

De esta forma hemos obtenido las 16 claves de 48 bits que usaremos en la red de Feistel. En el gráfico 4.5.2.4 podemos observar la obtención de las claves (K1….K16) en forma más amplia.

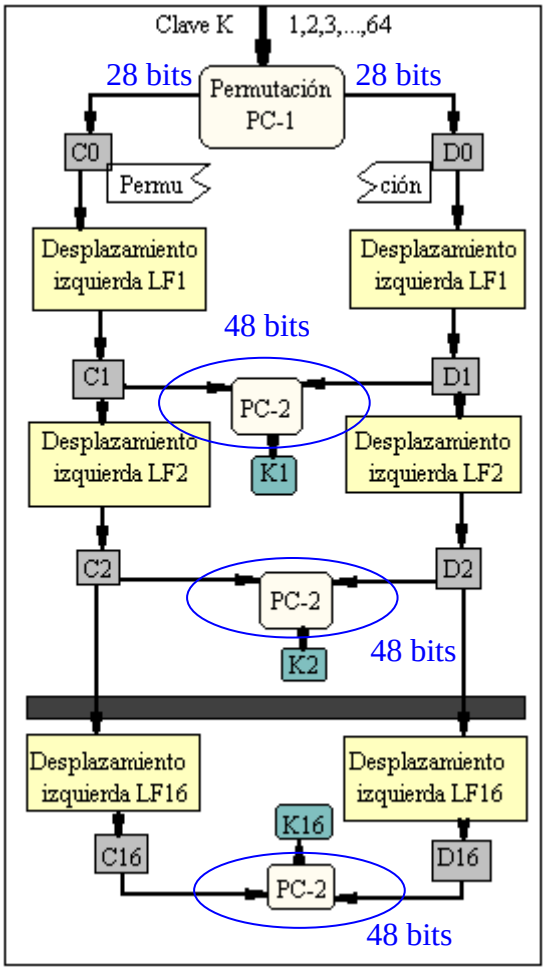
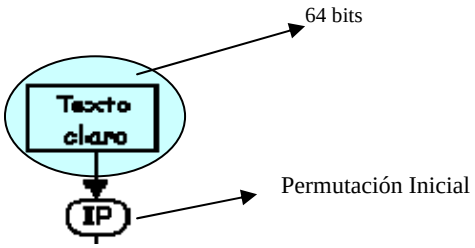


Figura 4.5.2.4 Cálculo de Subclaves en DES

Ahora vamos a cifrar un bloque de 64 bits. Lo primero que se hace es aplicarle una permutación normal IP *(Ver figura 4.5.2.5)* (ni de expansión ni de compresión), con lo que obtenemos el bloque permutado.



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

Li

Ri

Figura 4.5.2.4 Matriz de Permutación Inicial

Ahora se parte el bloque en dos mitades, Li y Ri como en la *figura 4.5.2.4*, entonces el orden de los valores que se obtienen para Li y Ri son:

Li = 58 50 42 34 26 18 10 02
60 52 44 36 28 20 12 04
62 54 46 38 30 22 14 06
64 56 48 40 32 24 16 08

Ri = 57 49 41 33 25 17 09 01
59 51 43 35 27 19 11 03
61 53 45 37 29 21 13 05
63 55 47 39 31 23 15 07

Luego de que el bloque haya pasado por la matriz de permutación entra a un ciclo que se repetirá durante N = 16, y lo destacamos a continuación:

- Se permuta la mitad derecha Ri aplicando expansión a 48 bits
- La clave de 56 bits se desplaza, permuta y se seleccionan los 48 bits de Ki
- La nueva mitad derecha Ri y la clave Ki se suman XOR
- Se reducen los 48 bits de salida a 32 bits mediante las Cajas-S
- Se permuta el resultado en la Caja-P
- El resultado se suma XOR con la mitad izquierda Li
- -Se intercambia de lugar Li y Ri

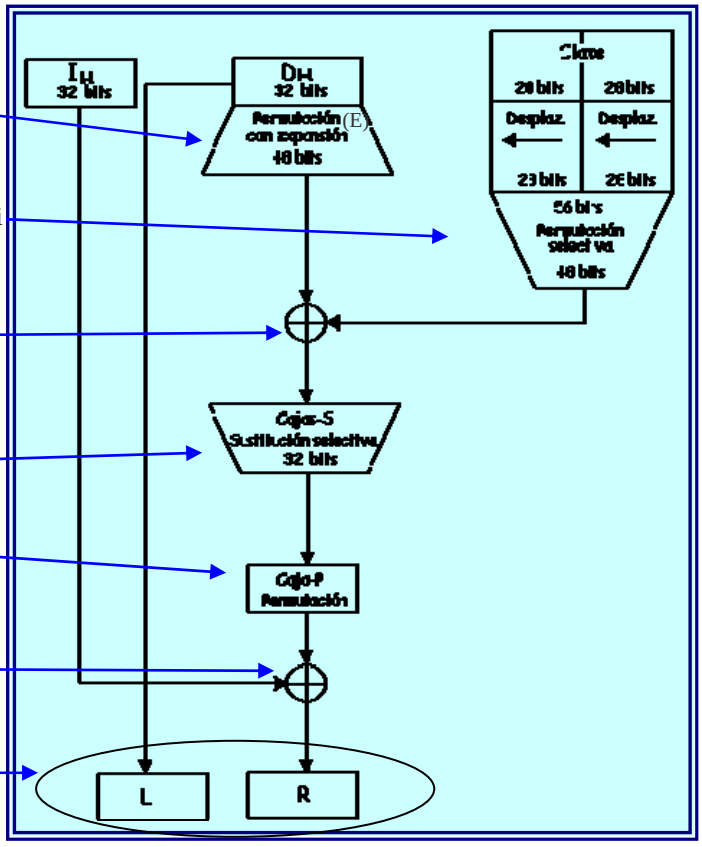
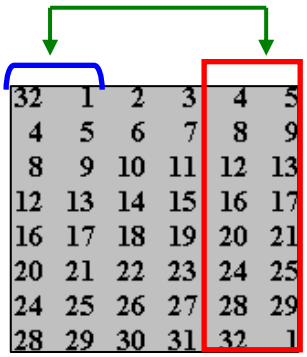


Figura 4.5.2.5 Ciclo de Operaciones en DES

Ahora trataremos de explicar el funcionamiento de las Cajas, las cuales como podemos observar en la *figura 4.5.2.5* son muy importantes dentro del algoritmo DES, y es en donde se podría presentar algún problema al momento de implementarlo.

La primera que se nos presenta es la de Permutación con expansión a 48 bits (**E**), estas realizan la tarea de tomar la palabra de 32 bits y aplicar una expansión 48 bits, el mismo largo que el de la clave (Ki).

Esto se realiza por la simple razón que necesitamos cadenas del mismo largo para poder aplicar el operador XOR. Esto se consigue repitiendo las dos primeras columnas recorridas una fila. *(Ver fig. 4.5.2.6)*



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

Figura 4.5.2.6 Tabla de Permutación (E) con Expansión

En cambio la Caja-P funciona como una simple matriz de permutación de 32 bits.

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

Figura 4.5.2.7 Tabla de Permutación (P) de 32 bits

Mientras que la Caja-S es donde radica la mayor dificultad para un criptoanálisis. Ya que esta representa a una función no lineal y unidireccional. *(Ver fig.4.5.2.8)*

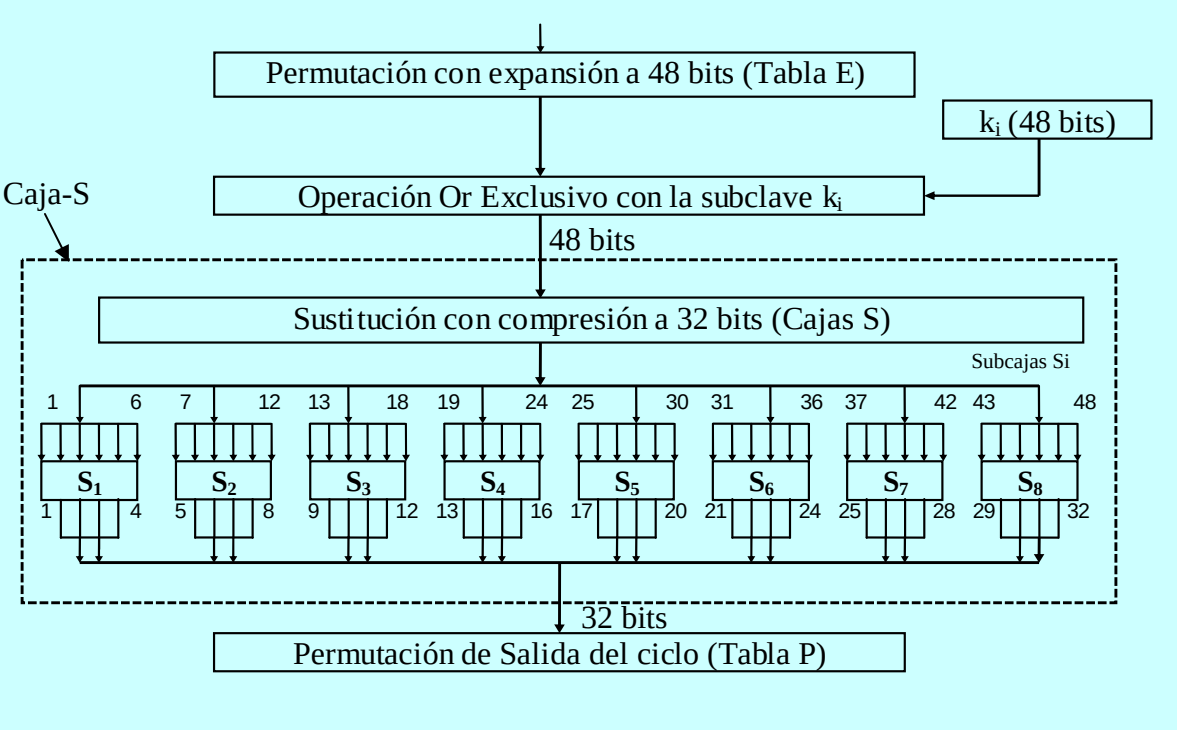


Figura 4.5.2.8 Caja-S

Ahora tenemos que de los 48 bits que entran como resultado del operador XOR entre la clave (Ki) y la tabla de permutación y expansión (E), estos 48 bits se dividen en 8 grupos de 6 bits cada uno, cada grupo ingresa a una subcaja denominada Si (S1,S2,S3...S8) que transforma los 6 bits de entrada en una salida de tan solo 4 bits. *(Ver fig.4.5.2.8)*

En el siguiente ejemplo trataremos de explicar el funcionamiento de la Caja-S y de las Subcajas Si que la conforman.

Ejemplo:

Tenemos los siguientes bits que están ubicados desde la posición 7 a la 12 dentro del bloque de 48 bits de entrada a la Caja-S. Los bits son los siguientes: **1 0 1 1 0 0.**

Como sabemos en que posición se encuentran los bits, entonces sabemos que corresponden como entrada de la Subcaja S2.

Para seleccionar la fila tomamos los bits extremos: $\mathbf{10_2=2_{10}= 2}$
Para seleccionar la columna tomamos los bits centrales: $\mathbf{0110_2 = 6_{10} = 6}$

De acuerdo a los valores obtenidos para la fila = 2 y la columna = 6 *(Ver fig. 4.5.2.9)*, la caja S2 indica una salida igual a

$$\mathbf{13 = 1101_2}$$

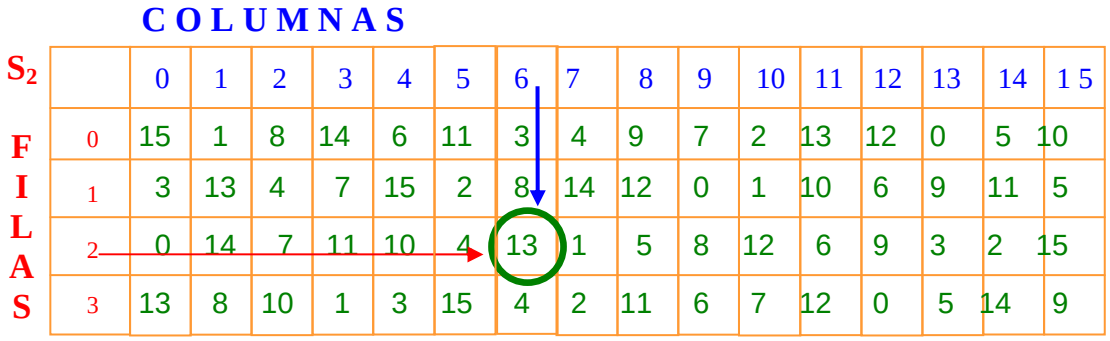


Figura. 4.5.2.9 Caja S_2

Resultado:

Para la entrada: **1 0 1 1 0 0** ⟶ **6 bits**

Tenemos la salida: **1 1 0 1** ⟶ **4 bits**

A continuación podemos observar los valores que corresponden a las 8 Subcajas del DES

COLUMNAS

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

COLUMNAS

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FILAS 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	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

COLUMNAS

S_3

FILAS	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	15	14	3	11	5	2	12

COLUMNAS

S_4

FILAS	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

COLUMNAS

S_5

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

COLUMNAS

S_6

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

En el último ciclo no hay que intercambiar las mitades.

Por último, luego de realizar los 16 ciclos, tenemos las dos mitades del principio, de 32 bits, ya cifradas. Las volvemos a unir, en un bloque de 64 bits, y las permutamos con la permutación inversa de la inicial IP^{-1}*(Ver fig 4.5.2.10).* El resultado es el bloque cifrado.

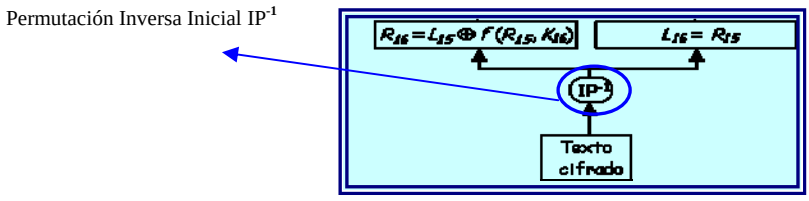


Figura 4.5.2.10 Permutación IP^{-1}

Para descifrar basta con utilizar el mismo algoritmo, empleando las claves Ki y los desplazamientos en orden inverso.

4.5.3.7 Claves Débiles y Semidébiles en DES

El algoritmo DES presenta algunas claves débiles. Todos los valores de la llave que conducen a una secuencia inadecuada de Ki serán poco recomendables. Distinguiremos entre claves débiles *(Ver fig. 4.5.3.1)*, que son aquellas que generan un conjunto de dieciséis valores iguales de Ki y que cumplen Ek(Ek(M)) = M, y claves semidébiles (cuadro 4.5.3.2), que generan dos valores diferentes de Ki, cada uno de los cuales aparece ocho veces.

En cualquier caso, el número de llaves de este tipo es tan pequeño en comparación con el número total de posibles claves, que no debe suponer un motivo de preocupación.

Clave	Clave tras aplicar PC1
0101010101010101	0000000 0000000
1F1F1F1F0E0E0E0E	0000000 FFFFFFF
E0E0E0E0F1F1F1F1	FFFFFFF 0000000
FEFEFEFEFEFEFEFE	FFFFFFF FFFFFFF

Figura 4.5.3.1 Claves Débiles en Hexadecimal

Clave	Clave tras aplicar PC1
01FE01FE01FE01FE	AAAAAAA AAAAAAA
FE01FE01FE01FE01	5555555 5555555
1FE01FE00EF10EF1	AAAAAAA 5555555
E01FE01FF10EF10E	5555555 AAAAAAA
01E001E001F101F1	AAAAAAA 0000000
E001E001F101F101	5555555 0000000
1FFE1FFE0EFE0EFE	AAAAAAA FFFFFFF
FE1FFE1FFE0EFE0E	5555555 FFFFFFF
011F011F010E010E	0000000 AAAAAAA
1F011F010E010E01	0000000 5555555
E0FEE0FEF1FEF1FE	FFFFFFF AAAAAAA
FEE0FEE0FEF1FEF1	FFFFFFF 5555555

Figura 4.5.3.2 Claves Semidébiles en Hexadecimal

4.5.4 Modos de Cifrado en DES

La función de cifrado puede ser de 4 tipos que los mencionaremos a continuación:

ECB: Electronic CodeBook (libro electrónico de códigos)
CBC: Cipher Block Chaining (encadenamiento de bloques)
CFB: Cipher FeedBack (realimentación de bloques)
OFB: Output FeedBack (realimentación bloque de salida)

4.5.4.1MODO ECB - Electronic CodeBook (Libro Electrónico de Códigos)

Cifra cada bloque con la clave k de forma independiente. Por lo tanto, el resultado es como si se codificase mediante un gran libro electrónico de códigos.

Debilidades:

- Se podría reconstruir ese libro electrónico sin necesidad de conocer la clave.
- Aparece el problema denominado de comienzos y finales fijos que permiten un tipo de ataque sencillo.
- Se ataca a través de la repetición de bloques similares.

Las principales características del modo ECB en DES es que cada bloque de 64 bits del texto en claro se pasa por el cifrador, usando la misma clave de 64 bits.

Para bloques de texto en claro iguales, se obtiene siempre el mismo criptograma.

Como a cada bloque de texto en claro le corresponde un único código o texto cifrado de salida y éste es constante, este modo de cifra lleva por nombre Libro Electrónico de Códigos. Es como si tuviésemos un gran libro de código con un código distinto para cada mensaje.

4.5.4.2Modo CBC - Cipher Block Chaining (Cifra por Encadenamiento de Bloques)

Se encadenan los bloques de texto en claro con el bloque del criptograma anterior, luego utiliza un vector de inicialización IV de 64 bits que se guarda en secreto.

Sus principales características son que evita el ataque por repetición de bloque; enmascara el mensaje lo mismo que la cifra en flujo; el espacio de claves es igual a 64 bits; la propagación de un error afecta a dos bloques contiguos.

El cifrado se realiza de la siguiente manera *(Ver fig.4.5.4.2):*

- El vector IV se suma XOR a los 64 bits de texto en claro.

- Se cifra con la clave K esa suma.
- El resultado Ci se usa como vector IV para el nuevo bloque.

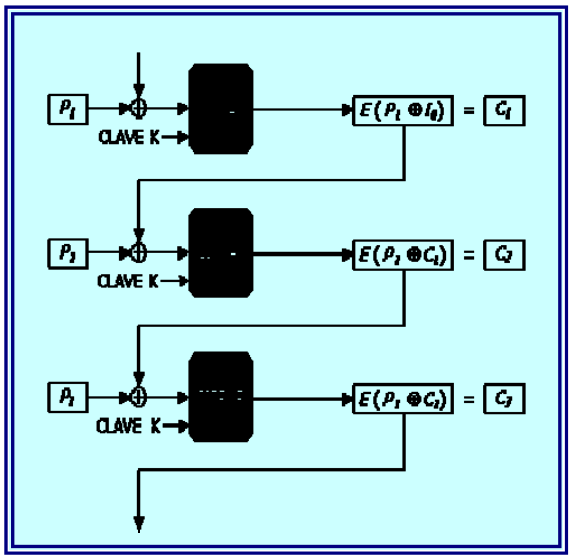


Figura.4.5.4.2 Cifrado CBC

Mientras que para descifrar se realiza de la siguiente manera:

- Se descifra el primer bloque con vector IV:
  - P1 = D(C1) ⊕ I0
  - P1 = D[E(P1⊕ I0)] ⊕ I0
- Se guarda el bloque Ci-1 en un registro.
- Se descifra el bloque Ci y luego XOR entre esos bloques:
  - Mi = D(Ci) ⊕ Ci-1

4.5.4.3 Modo CFB - Cipher Feedback (Cifra por Realimentación de Bloques)

- Se pueden cifrar unidades de datos más pequeñas que bloques, por lo general un byte.
- Se usa un registro de desplazamiento RD de 64 bits como vector inicial IV.

Las características que nos presenta este cifrado son; evita el ataque por repetición de bloque; enmascara el mensaje como en cifra en flujo, el espacio de claves es igual a 64 bits; la propagación de un error se limita a un bloque.

Para cifrar por realimentación de bloques seguimos los siguientes pasos *(Ver fig. 4.5.4.3)*

- Se suma XOR cada byte del texto claro con bytes resultado de la cifra de RD y la clave K.
- El byte Ci se envía al registro; se desplaza 8 bits a la izquierda hasta formar otro RD y se repite el proceso de cifra.

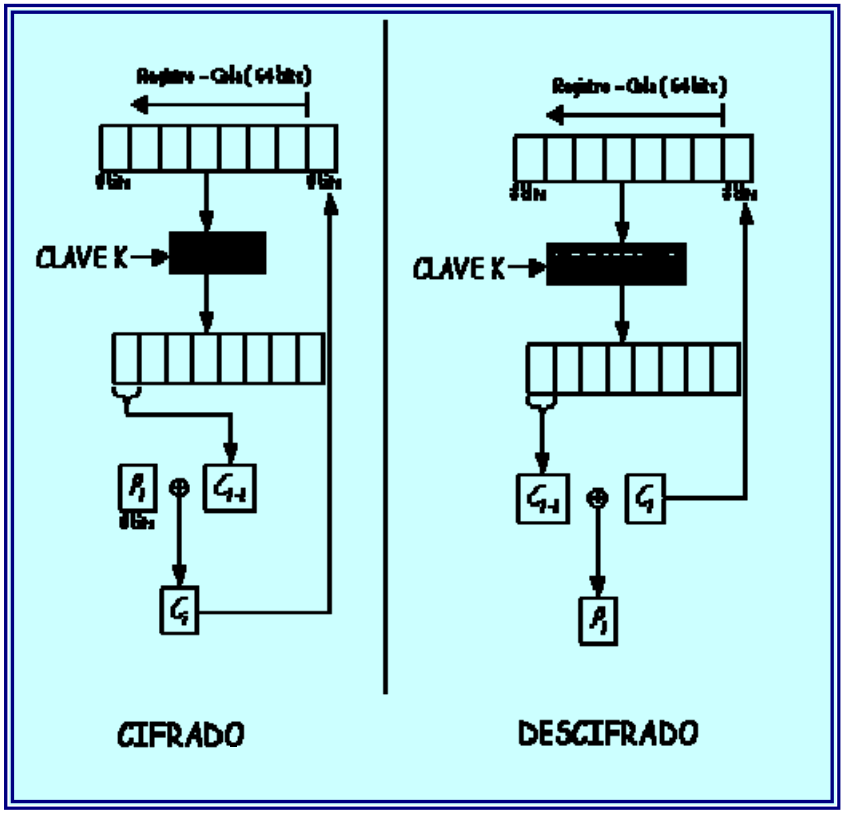


Figura 4.5.4.3 Cifrado CFB

Para descifrar procedemos de la siguiente manera:

- Se cifra el registro RD.
- Se obtienen de esta forma los elementos de Ci-d.
- Se suma XOR los Ci-d con los Ci del criptograma para obtener Pi.
- Se realimenta Ci al registro RD y se repite el proceso.

4.5.4.4 Modo OFB - Output Feedback (Cifra por realimentación de bloques de salida)

La realimentación de la señal se realiza antes de la operación XOR.
El DES, la clave y el Registro RD actúan como un generador de secuencia cifrante. *(Ver fig4.5.4.4)*

Sus principales características son las siguientes:

- Evita el ataque por repetición de bloque.
- Produce un enmascaramiento del mensaje similar al de un cifrador de flujo.
- El espacio de claves es igual a 64 bits.
- La propagación de un error afecta sólo a un byte, el que se realimenta en el registro de desplazamiento.
- Las operaciones de cifrado y descifrado son iguales.
- Si la cifra se realiza bit a bit, OFB se convierte en cifrador de flujo.

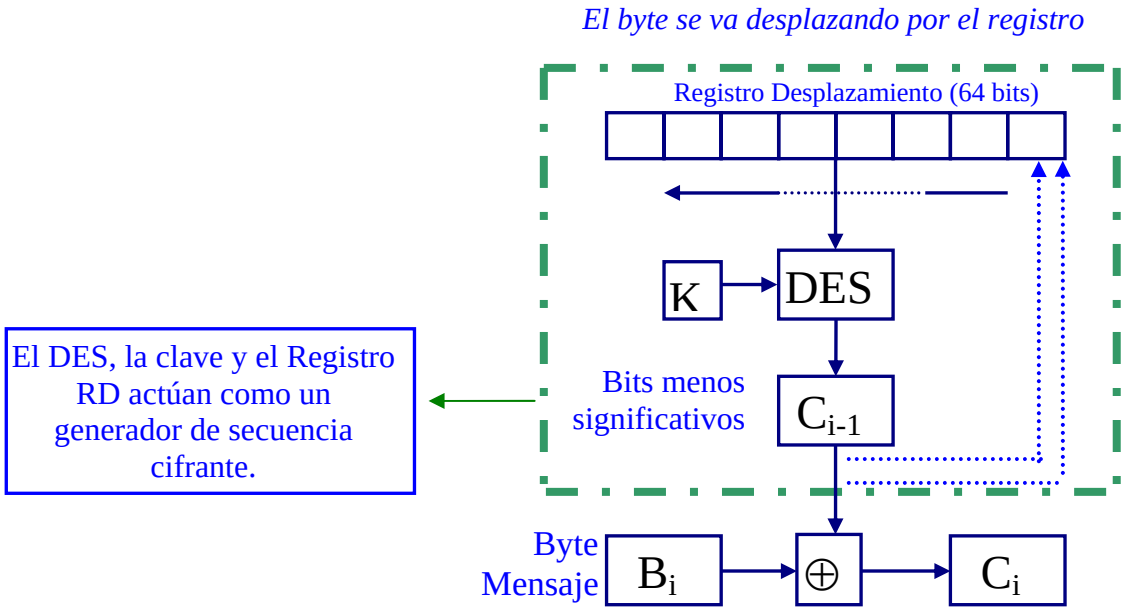


Figura 4.5.4.4 Cifrado OFB

4.5.5 Criptosistema de Triple DES

Esta es una variación del criptosistema DES tradicional aunque es mucho más seguro tanto que tiene un valor efectivo de longitud de clave igual a 2^{2n} bits, es decir $2^{2 \cdot 56} = 2112$ bits.

Además es inmune a ataques de encuentro a medio camino. Funciona de la siguiente manera *(Ver figura 4.5.5):*

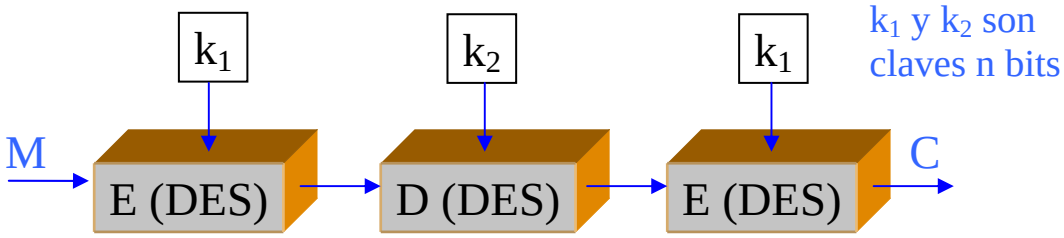


Figura 4.5.5 Triple DES

En este caso las dos claves funcionan como 3 claves.

El proceso comienza con cifrar mediante el método DES con una clave k1, luego se desencripta con una clave k2 y por último se vuelve a cifrar con la clave k1.

A pesar que en la actualidad los ataques han logrado derrotar al sistema DES por lo cual este no a sido certificado nuevamente por la NIST, pero el sistema de triple DES sigue siendo seguro y es aplicado en varios programas como puede ser PGP.

4.6 Criptografía de Clave Pública o Asimétrica

Los algoritmos de llave pública, o algoritmos asimétricos, han demostrado su interés para ser empleados en redes de comunicación inseguras (Internet). Introducidos por Whitfield Diffie y Martin Hellman a mediados de los años 70, su novedad fundamental con respecto a la criptografía de clave secreta es que las claves no son únicas, sino que forman pares.

Hasta la fecha han aparecido multitud de algoritmos asimétricos o de clave publica, la mayoría de los cuales son inseguros; otros son poco prácticos, bien sea porque el criptograma es considerablemente mayor que el mensaje original, bien sea porque la longitud de la clave es enorme. Se basan en general en plantear al atacante problemas matemáticos difíciles de resolver.

En la práctica muy pocos algoritmos son realmente útiles. El más popular por su sencillez es RSA, que ha sobrevivido a multitud de ataques, si bien necesita una longitud de clave considerable. Otros algoritmos son los de ElGamal y Rabin.

Los algoritmos de clave publica emplean generalmente longitudes de clave mucho mayores que los de clave privada. Por ejemplo, mientras que para algoritmos simétricos se considera segura una clave de 128 bits, para algoritmos asimétricos se recomiendan claves de al menos 1024 bits. Además, la complejidad de cálculo de estos últimos los hace considerablemente más lentos que los algoritmos de cifrado simétricos. En la práctica los métodos asimétricos se emplean únicamente para codificar la clave de sesión (simétrica) de cada mensaje o transacción particular.

Los algoritmos asimétricos o de clave pública poseen dos claves diferentes en lugar de una, Kp y KP, denominadas clave privada y clave pública. Una de ellas se emplea para codificar, mientras que la otra se usa para decodificar. Dependiendo de la aplicación que le demos al algoritmo, la clave pública sería la de cifrado o viceversa. Para que estos criptosistemas sean seguros también debe cumplirse que a partir de una de las claves resulte extremadamente difícil calcular la otra.

Una de las aplicaciones inmediatas de los algoritmos asimétricos es el cifrado de la información sin tener que transmitir la clave de decodificación, lo cual permite su uso en canales inseguros. Supongamos que A quiere enviar un mensaje a B *(Ver Fig. 4.6)*. Para ello solicita a B su clave pública KP . A genera entonces el mensaje cifrado EKP (m). Una vez hecho esto únicamente quien posea la clave Kp, o sea el receptor B podrá recuperar el mensaje original m.

Nótese que para este tipo de aplicación, la llave que se hace pública es aquella que permite codificar los mensajes, mientras que la llave privada es aquella que permite descifrarlos.

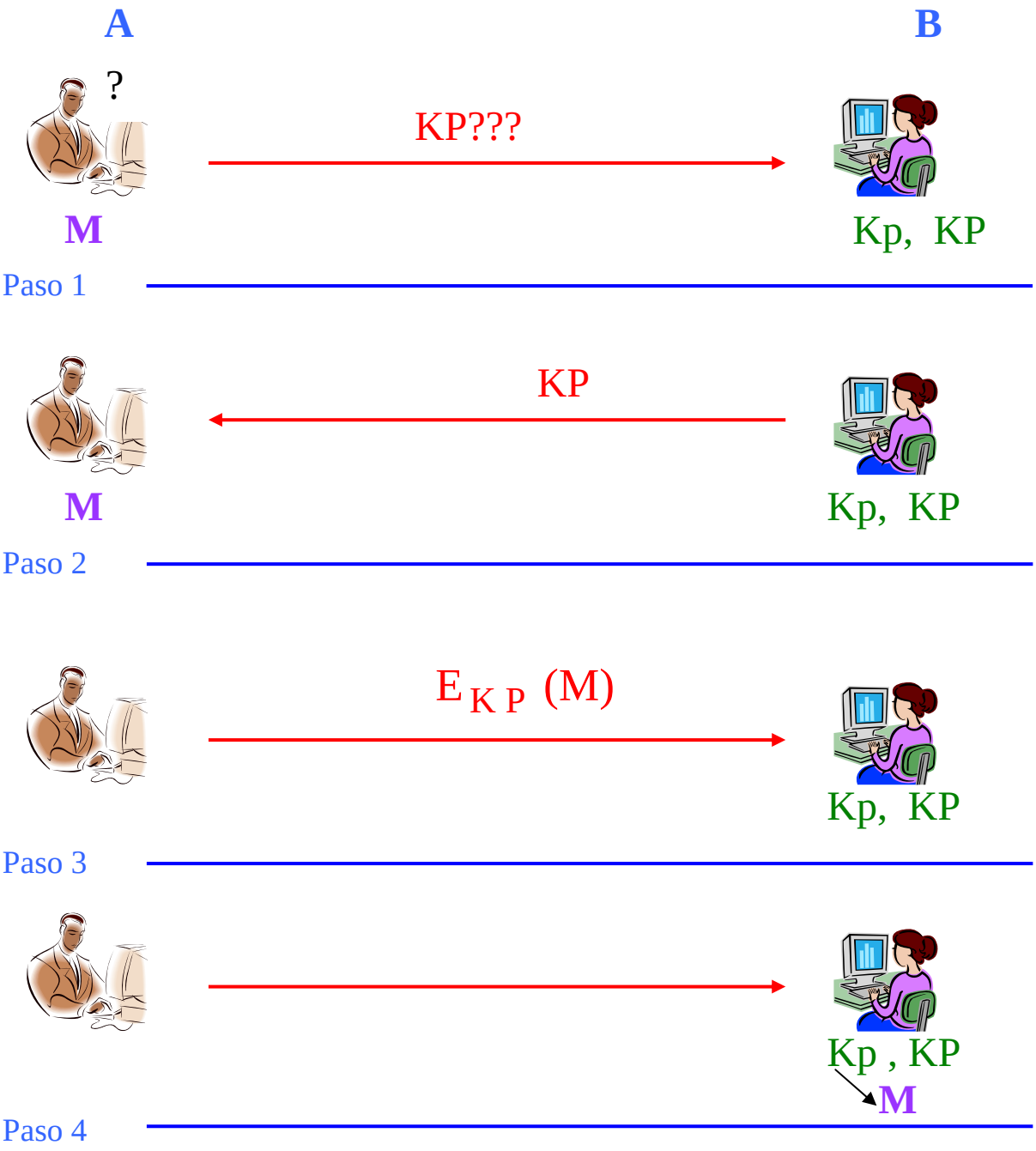


Figura 4.6.1 Transmisión de la información

Ahora vamos a explicar el gráfico paso a paso

Paso 1:	El usuario **A** desea enviar el mensaje **M** a **B**, pero no conoce su clave pública, por lo cual solicita la clave pública de **B**
Paso 2:	La usuario **B** le contesta a **A** enviándole su clave pública **KP**.

Paso 3: **A** codifica el mensaje **M** y envía a **B** el criptograma $\mathbf{E_{KP}\ (M).}$

Paso 4: **B** decodifica el criptograma $\mathbf{E_{KP}\ (M)}$ empleando la clave privada **Kp**.

Más adelante estudiaremos a fondo las diversas aplicaciones de la criptografía, como por ejemplo la Autentificación, las Firmas Digitales, etc.

4.6.1 Criptosistema RSA

La fortaleza de este sistema radica en la dificultad de la factorización de números primos granes, dicho de otra forma, que si queremos descubrir una clave privada a partir de la clave publica nos enfrentaremos a un problema que computacionalmente es intratable, por el tiempo de procesamiento que tendría y talvez cuando el ataque tenga resultado ya se haya cambiado la clave o la información ya no sea de utilidad.

Este sistema consta de una clave pública y una privada, las cuales están conformadas por dos números, en el caso de la clave pública esta conformada de un número "e" que es un número elegido dentro de un cuerpo finito, y un número "n" que es el cuerpo del cual se codificara.

Por otro lado la clave privada consta de un número "d" que es el inverso de "e" en el mismo cuerpo, y además el mismo "n".

Una vez que se tiene el par de claves el cifrado es muy sencillo, sigue una fórmula demasiado sencilla que es:

$$C = E_{eR}(M) = M^{eR} \bmod n_R$$

- Donde 'C' es el criptograma.
- 'M' es el mensaje en claro, y debe ser un elemento de CCR del cuerpo finito.
- 'n' el cuerpo finito en donde se representa el criptograma.
- 'E' función de encriptación.
- 'e' número primo que pertenece a la clave publica.

En cambio para descifrar se utiliza la formula:

$$M = E_{dR}(C) = C^{dR} \bmod n_R$$

Dado que ‘e’ es un valor relativamente pequeño su inverso puede ser muy grande por los que este cálculo de descifrado puede ser muy pesado, generando números demasiado grandes y de difícil manejo, es posible resolver esto mediante otro método denominado “Teorema del resto Chino”.

$$M = \{A_p[C_p^{\,dp} \pmod p] + A_q[C_q^{\,dq} \pmod q]\} \bmod$$

$$\text{con} \quad C_p = C \bmod p \qquad C_q = C \bmod q$$

$$d_p = d \bmod (p-1) \qquad d_q = d \bmod (q-1)$$

$$A_p = q\,[\text{inv}\,(q,p)] = q^{p-1} \bmod n$$

$$A_q = p\,[\text{inv}\,(p,q)] = p^{q-1} \bmod n$$

De esta forma parece más complicada que la simple potencia en módulo de ‘n’ pero computacionalmente es mucho más rápida y genera números mucho más pequeños y tratables. Además al ser el dueño y generador de la clave privada, el que descifre con este sistema, es también es el que conoce la trampa de ‘p’ y ‘q’ que son dos números primos con los que se genera el para de claves.

Una vez conocida la parte sencilla del sistema RSA continuaremos con la parte difícil del sistema, esta es la generación de la clave.

El primer paso es la selección de nos números primos, pero el sistema necesita que estos dos números primos (p, q) cumplan con ciertas condiciones.

- La primera condición es que sean primos seguros, esto quiere decir que son primos que cumplen con p=2*p’+1 donde “p” es el primo seguro y p’ un número primo.
- “p” y ”q” deben diferir en pocos dígitos pero no deben ser muy cercanos.
- La longitud de “p” y “q” debe ser de mínimo 500 bits
- Valores de (p-1) y (q-1) del Indicador de Euler deben tener factores primos grandes.
- El mcd entre p-1 y q-1 debe ser pequeño (de preferencia 2).
- Elegimos enteros $x > \sqrt{n}$ hasta que $(x^2 - n)$ sea cuadrado perfecto. En este caso x = (p+q)/2; y = (p-q)/2. Por lo tanto rompemos el valor n: p = (x+y); q = (x-y) **nota: “n” es el resultado de la multiplicación de p*q=n.**

Se elige p' un primo grande de modo que :p = 2∗p' + 1

Se elige q' un primo grande de modo que: q = 2∗q' + 1

Vamos a poner un pequeño ejemplo para que quede un poco más claro:

Si tenemos a p' = 1.019 y q' = 3.863

p = 2∗1.019 + 1 = 2.039 (11 bits) **Es primo**
q = 2∗3.863 + 1 = 7.727 (13 bits) **Es primo**
n = p∗q = 15.755.353

Entonces: p-1 = 2.038; q-1 = 7.726

p-1 = 2∗1.019; q-1 = 2∗3.863 $\Rightarrow$ mcd (p-1, q-1) = 2

Los primos p y q cumplen la condición de **primos seguros**

Una mala elección de estos números podría llevar a facilitar o por lo menos que lleve menos tiempo el criptoanálisis de un mensaje.

Aparte de d como clave privada, existen claves privadas parejas, esto quiere decir que existen más números dentro del cuerpo finito que actúan como d (que al aplicarlos para el descifrado nos dan como resultado el mensaje en claro. Esto se puede calcular mediante la formula:

Si: γ = mcm (p-1), (q-1) sea $d\gamma$ = e-1 mod γ = inv (e, γ)

La clave pública e tendrá λ claves parejas de forma que:

$di = d\gamma + i\,\gamma$ $\quad$ $1 < di < n$

$i = 0, 1, \ldots \lambda$ $\quad$ $\lambda = \lfloor (n - d\gamma)/\rfloor\gamma$

En el siguiente ejemplo podemos observar los problemas que se nos presenta al momento de encontrar las claves privadas parejas.

Si p = 7; q = 19; n = 133, $\phi(n)$ = 108; elegimos e = 11.

Calculamos d = inv (11, 108) = 59

Si ciframos el mensaje M = 60,

Obtenemos $C = 60^{11} \bmod 133 = 86$.

Luego desciframos $C^d \bmod n = 86^{59} \bmod 133 = 60$

Pero también con d' = 5, 23, 41, 77, 95, 113, 131

En el ejemplo anterior tenemos que:

$\gamma = mcm\ (6, 18) = 18.$

Luego: $d_\gamma = inv\ (11, 18) = 5,$

Así $d_i = d_\gamma + i\ \gamma = 5 + i*18.$

Es decir $d_i = 5, 23, 41, 59, 77; 95, 113, 131.$

Están las 7 claves parejas $\lambda = \lfloor (n - d_\gamma)/\gamma \rfloor = \lfloor (133 - 5)/18) \rfloor = 7$, más la clave privada 59.

En este caso hablamos de debilidad de la clave, al igual que sucedía con las claves débiles y semidébiles por en DES.

El siguiente paso después de la elección de los números 'p' y 'q' y además 'n' es calcular el valor de $\phi(n) = (p-1)(q-1)$ –teorema de Euler- este valor nos servirá como parámetro para la elección de 'e'que con 'n' conformaran la clave publica.

El valor de 'e' debe cumplir con las siguiente condición: Se elige una clave pública e, tal que $1 < e < \phi(n)$ y que cumpla con $mcd\ [e, \phi(n)] = 1$.

En este punto nos enfrentamos con otro problema, el de los mensajes no cifrables, con esto nos referimos a mensajes que pasan por le algoritmo de cifrado y como resultado obtenemos el mismo mensaje, el número de mensajes no cifrables esta dado por las siguientes ecuaciones.

El número de mensajes no cifrables dentro de un cuerpo n viene dado por:

$$\sigma_n = [1 + mcd\ (e-1, p-1)][1 + mcd\ (e-1, q-1)]$$

Los mensajes no cifrables serán:

$$M = [q\{inv\ (q, p)\}M_p + p\{inv\ (p, q)\}M_q] \bmod n$$

con: M_p las soluciones de $M^e \bmod p = M$

M_q las soluciones de $M^e \bmod q = M$

Esto último debido al TRC puesto que $M^e \bmod n = M$

para minimizar el número de mensajes no cifrables se debe elegir una clave 'e' de tal forma que:

$$mcd\ (e-1, p-1) = 2 \quad y \quad mcd\ (e-1, q-1) = 2$$

Entonces: $\sigma_n = [1 + 2][1 + 2] = 9$

Esto se logra usando primos seguros:

$p = 2p' + 1$ y $q = 2q' + 1$ con p' y q' primos grandes

ya que:

mcd (e-1, p-1) = mcd (e-1, (2p' +1)-1) $\Rightarrow$ mcd = 2 o bien p'
mcd (e-1, q-1) = mcd (e-1, (2q' +1)-1) $\Rightarrow$ mcd = 2 o bien q'

Luego: $\sigma_n = \{9, 3(p'+1), 3(q'+1), (p'+1)(q'+1)\}$

Hay que comprobar en diseño que no se den valores del mcd igual a p' o q' pues tendríamos un número muy alto de estos mensajes.

Vale aclarar que el número mínimo de mensajes no cifrable es de 9 por que es le número menor que puede resultar de la ecuación .

Después de haber elegido el número 'e' se procede a elegir el número 'd' que es el inverso de 'e' dentro del cuerpo $\phi(n)$ -d = inv [e,$\phi(n)$]-.

Como último paso del algoritmo de RSA se hace publico 'e' y 'n' que conformaran la clave publica. Y se mantendrán secretos los números 'd', 'p' y 'q'. En este punto nos preguntaremos porque 'p' y 'q' y no destruirlos y mantener 'n', es posible destruir 'p' y 'q' pero estos nos servirán para realizar el descifrado por medio del teorema chino y si son destruidos el sistema gana un poco de seguridad pero se volvería más lento, esto último es un recordatorio de lo ya dicho en la explicación del descifrado.

4.6.1.1 Ejemplo de descifrado del Criptosistema RSA

A continuación un ejemplo del descifrado RSA aplicando el Teorema del Resto Chino (TRC) en n,

$M = Cd \bmod n$

Sea: $p = 89$, $q = 31$, $n = p*q = 89*31 = 2.759$, $\phi(n) = 88*30 = 2.640$

Elegimos $e = 29 \Rightarrow d = \text{inv}[e, \phi(n)] = \text{inv}[29, 2.640] = 2.549$

Si el bloque de mensaje ya codificado es M = 1.995, entonces:

$C = M^e \bmod n = 1.995^{29} \bmod 2.759 = 141$

$M = C^d \bmod n = 141^{2.549} \bmod 2.759 = \mathbf{1.995}$

$Ap = q^{p-1} \bmod n = 31^{88} \bmod 2.759 = 713$

$Cp = C \bmod p = 141 \bmod 89 = 52$

$dp = d \bmod (p-1) = 2.549 \bmod 88 = 85$

$Aq = p^{q-1} \bmod n = 89^{30} \bmod 2.759 = 2.047$

$Cq = C \bmod q = 141 \bmod 31 = 17$

$dq = d \bmod (q-1) = 2.549 \bmod 30 = 29$

Reemplazando en:

$M = \{A_p[C_p dp \ (\bmod p)] + A_q[C_q^{dq} \ (\bmod q)]\} \bmod n$

$M = \{713[52^{85} \bmod 89] + 2.047[17^{29} \bmod 31]\} \bmod 2.759$

$M = \{713*37 + 2.047*11\} \bmod 2.759 = (26.381 + 22.517) \bmod 2.759 =$ **1.995**

En este ejemplo se muestra el descifrado en sus dos formas por una forma sencilla lógicamente pero casi intratable computacionalmente, esto por lo grande de los números resultantes de la operación de exponenciación, y una secunda por medio del "Teorema del resto chino" que es un poco más complicada lógicamente pero genera números pequeños que computacionalmente son tratables. Vale recalcar que este ejemplo es con números pequeños, y en la realidad estos cálculos se realizan con números mayores a 500 bits con lo que son números extremadamente grandes.

4.6.2 Criptosistema ElGamal

Este sistema esta basado en que el atacante tiene que enfrentarse al problema del Logaritmo Discreto de un número grande. Este problema también es de un tiempo de ejecución no polinominal.

Las características de este algoritmo son:

- Se elige un grupo multiplicativo Z_p^*, p es primo grande.
- Del grupo p se elige una raízα, generador del grupo.
- Cada usuario elige un número aleatorio λ dentro de p.
  - Esta será la clave privada.
- Cada usuario calcula $\alpha^{\lambda} \bmod p$.
  - Junto con p es la clave pública.

El algoritmo funciona de la siguiente manera:

Operación Cifrado: A cifra un mensaje M que envía a B

- El usuario B ha elegido su clave privada *b* dentro del cuerpo del número primo p que es público.
- El usuario B ha hecho pública su clave $\alpha^b \bmod p$.
- El emisor A genera un número aleatorio ν de sesión y calcula $\alpha^\nu \bmod p$.
- Con la clave pública de B (α^b) el emisor A calcula:
- $(\alpha^b)^\nu \bmod p$ y $M*(\alpha^b)^\nu \bmod p$
- A envía a B el par: $C = [\alpha^\nu \bmod p, M * (\alpha^b)^\nu \bmod p]$

Operación Descifrado: B descifra el criptograma C que envía A

- El usuario B recibe $C = [\alpha^\nu \bmod p, M * (\alpha^b)^\nu \bmod p]$.
- B toma el valor $\alpha^\nu \bmod p$ y calcula $(\alpha^\nu)^b \bmod p$.
- B descifra el criptograma C haciendo la siguiente división:
- $[M * (\alpha^b)^\nu \bmod p\ ] / [(\alpha^\nu)^b \bmod p]$ **$(\alpha\wedge^{\mathbf{b}})\wedge^{\boldsymbol{\nu}} = (\alpha\wedge^{\boldsymbol{\nu}})\wedge^{\mathbf{b}}$**

El paso anterior es posible hacerlo porque existirá el inverso de $(\alpha^\nu)^b$ en el grupo p al ser p un primo.

Luego:
$[M * (\alpha^b)^\nu * \{\text{inv}\ (\alpha^\nu)^b, p\}] \bmod p = M.$

4.6.2.1Ejemplo Criptosistema ElGamal

A continuación presentamos un ejemplo de este cifrado.
Cifrado.

Claves públicas de Benito: $p = 13$, $\alpha = 6$, $(\alpha^b) \bmod p = 2$

Adela A elige por ejemplo $\nu = 4$ y calcula:

$(\alpha^\nu) \bmod p = 64 \bmod 13 = 9$
$(\alpha^b)^\nu \bmod p = 24 \bmod 13 = 3$
$M*(\alpha^b)^\nu \bmod p = 10*3 \bmod 13 = 4$

Envía a B $(\alpha^\nu) \bmod p$, $M*(\alpha^b)^\nu \bmod p = [9, 4]$

Descifrado.

La clave privada de Benito es b = 5
Benito recibe: $[(\alpha^\nu) \bmod p, M*(\alpha^b)^\nu \bmod p] = [9, 4]$
Benito calcula:

$(\alpha^\nu)^b \bmod p = 95 \bmod 13 = 3$
$[M*(\alpha^b)^\nu] * \text{inv}[(\alpha^\nu)^b, p] = 4 * \text{inv}\ (3, 13) = 4 * 9$

$M = 4 * 9 \bmod 13 = 10$ (se recupera el mensaje).

4.6.3 Consideraciones sobre el Cifrado de Clave Pública

Para cifrar los mensajes se necesita que el mensaje M se transforma en números y éstos se dividen en bloques de g-1 dígitos, siendo g el número de dígitos del módulo de trabajo: el valor n = n∗p para RSA y p para ElGamal. En la práctica esto no ocurrirá ya que el cuerpo de cifra es como mínimo de 512 bits y el "mensaje" a cifrar tendrá sólo una centena de bits.

Se representará el mensaje en su valor ANSI decimal.

n = p∗q = 89∗127 = 11.303 ⇒ bloques de cuatro dígitos
$\phi(n)$ = 11.088; e = 25; d = inv (25, 11.088) = 10.201
M = Olé = 079- 108- 233 ⇒ M = 0791- 0823- **3**

Cifrado

$C1 = 791^{25} \bmod 11.303 = 7.853$
$C2 = 823^{25} \bmod 11.303 = 2.460$
$C3 = 3^{25} \bmod 11.303 = 6.970$

Descifrado

$M1 = 7.853^{10201} \bmod 11.303 = 0791$
$M2 = 2.460^{10201} \bmod 11.303 = 0823$
$M3 = 6.970^{10201} \bmod 11.303 = \mathbf{3}$

Además otra consideración sobre estos algoritmos esta en la complejidad de los problemas a los que los atacantes se enfrentan.

El problema de la Factorización de Números Grandes PFNG tiene una complejidad similar al del Logaritmo Discreto PLD: ambos suponen un tiempo de ejecución no polinomial.

Número de pasos que debe realizar el algoritmo PFNG:

$e\sqrt{\{\ln(n)*\ln[\ln(n)]\}}$

n = 60 dígitos ⇒ 2,7∗1011 pasos
n = 100 dígitos ⇒ 2,3∗1015 pasos
n = 200 dígitos ⇒ 1,2∗1023 pasos

El PLD es matemáticamente similar:

número de pasos ≈ $\mathbf{e\sqrt{\{\ln(p)*\ln[\ln(p)]\}}}$

4.7 Funciones Hash en Criptografía

Dado que los sistemas de clave pública son muy lentos, en vez de firmar digitalmente el mensaje completo, en un sistema criptográfico se incluirá como firma digital una operación con la clave privada sobre un resumen o hash de dicho mensaje representado por sólo una centena de bits.

4.7.1 Seguridad Asociadas a una Función Hash.

La seguridad que brindan las funciones de Hash esta principalmente en su aplicación para mantener integro un mensaje. Con esta función se pretende comprobar que durante el viaje que realiza el mensaje desde el emisor hasta el receptor, este no sea alterado por un tercero.

Esto se realiza mediante el siguiente funcionamiento.
Mensaje = M $\Rightarrow$ Función Resumen = h(M)
Firma (rúbrica): $r = E_{Kp}\{h(M)\}$

dE es la clave privada del emisor que firmará h(M).

Y para comprobar la identidad del emisor, el receptor descifra la rúbrica r con la clave pública del emisor KP. Al mensaje en claro recibido M' (se descifra si viene cifrado) se le aplica la misma función hash que en emisión. Si los valores son iguales, la firma es auténtica y el mensaje íntegro:

Calcula:	$E_{KP}(r) = h(M)$
Compara:	¿h(M') = h(M)?

La función de resumen puede ser de unos pocos bits, un ejemplo de esto son los algoritmos más utilizados, MD5 que resume a 128 bits y SHA-1 que resume a 160 bits.

A pesar que SHA-1 es mucho más lento que MD5, es más utilizado por su mayor seguridad que esta dada por la dificultad de generar dos mensajes aleatorios distintos con el mismo resumen es en MD5 es de 2^{60} mientras que para SHA-1 es de 2^{80}.

4.7.2 Propiedades de las Funciones Hash.

Las propiedades de las funciones hash, para que esta pueda considerarse segura deben ser:

- **Unidireccionalidad.** Conocido un resumen h(M), debe ser computacionalmente imposible encontrar M a partir de dicho resumen.

- **Compresión**. A partir de un mensaje de cualquier longitud, el resumen h(M) debe tener una longitud fija. Lo normal es que la longitud de h(M) sea menor.
- **Facilidad de cálculo.** Debe ser fácil calcular h(M) a partir de un mensaje M.
- **Difusión.** El resumen h(M) debe ser una función compleja de todos los bits del mensaje M. Si se modifica un bit del mensaje M, el hash h(M) debería cambiar aproximadamente la mitad de sus bits.
- **Colisión simple**. Conocido M, será computacionalmente imposible encontrar otro M' tal que h(M) = h(M'). Se conoce como *resistencia débil a las colisiones*.
- **Colisión fuerte**. Será computacionalmente difícil encontrar un par (M, M') de forma que h(M) = h(M'). Se conoce como *resistencia fuerte a las colisiones*.

4.8 Aplicaciones Criptográficas

Una vez que hemos logrado estudiar diversos aspectos acerca del cifrado de la información, podemos también analizar otras aplicaciones criptográficas tales la autenticación de mensajes con la ayuda de funciones resumen (hash) lo que deriva en las famosas firmas digitales.

4.8.1 Autenticación

Por autentificación entenderemos cualquier método que nos permita comprobar de manera segura alguna característica sobre un objeto. Dicha característica puede ser su origen, su integridad, su identidad, etc.

Un usuario que maneje la información por lo general desea garantizar la procedencia de un mensaje conocido, de forma que este pueda asegurarse de que no es una falsificación. Este mecanismo se conoce habitualmente como firma digital.

4.8.2 Firmas Digitales

Permiten al receptor verificar la autenticidad del origen de la información, así como también que la información esta intacta (integridad) y el No-Repudio o sea evitar que el emisor argumente que no envío la información.

Las firmas digitales tienen el mismo propósito que el de una firma escrita. Los principales requisitos que se deben tener en cuenta para generar una firma digital son las siguientes:

- Debe ser fácil de generar.
- Será única, sólo posible de generar por su propietario.
- Será fácil de autenticar o reconocer por su propietario y los usuarios receptores.

- Debe depender del mensaje y del autor.

Son condiciones más fuertes que la de una firma manuscrita, pero tienen la ventaja de que no pueden ser falsificadas tan fácilmente como las escritas.

Existen diferentes modos de implementación, como por ejemplo cifrar con la clave privada todo el mensaje **M** pero este es muy lento generando la firma *(Ver fig. 4.8.2 a).*

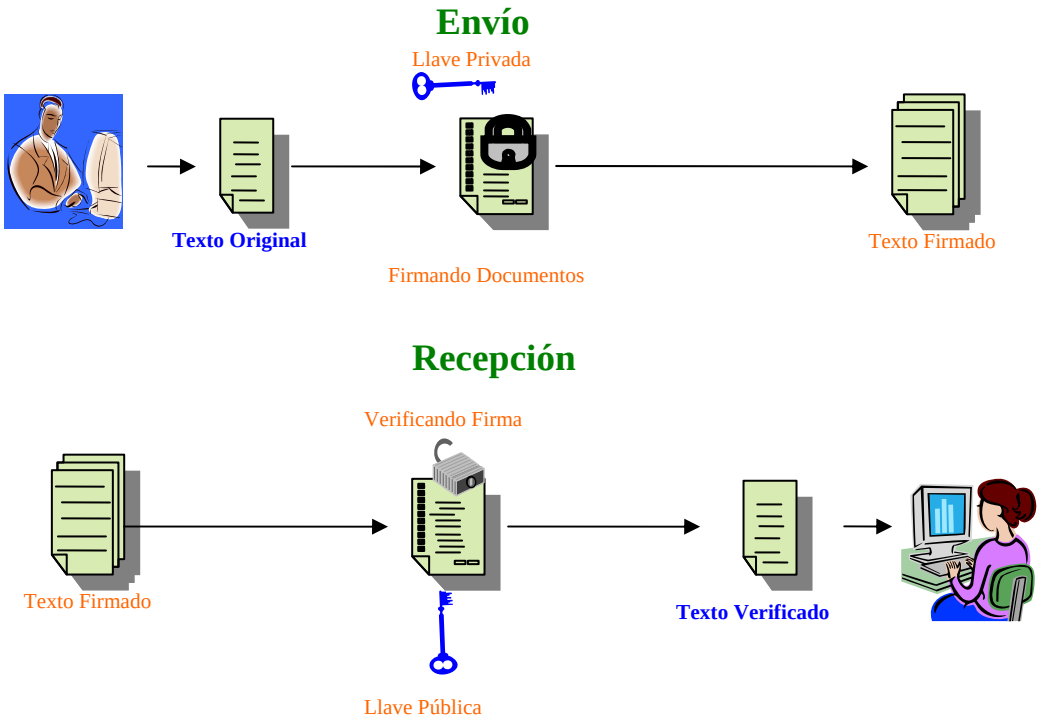


Figura 4.8.2 a. ***Firma Digital del mensaje completo***

Como también podría hacerse un resumen digital de **M** y una clave privada no codificando el mensaje completo. Estas funciones resumen, también conocidas como MDC (modification detection codes), nos van a permitir crear firmas digitales.

Sabemos que un mensaje m puede ser autentificado codificando con la llave privada Kp el resultado de aplicarle una función resumen, EKp(r(m)). Esa información adicional (que denominaremos firma del mensaje m) sólo puede ser generada por el poseedor de la clave privada Kp. Cualquiera que tenga la llave pública correspondiente estará en condiciones de decodificar y verificar la firma *(Ver figura 4.8.2.b)*. Para que sea segura, la función resumen r(x) debe cumplir además ciertas características:

- r(m) es de longitud fija, independientemente de la longitud de m.
- Dado m, es fácil calcular r(m).
- Dado r(m), es computacionalmente intratable recuperar m.
- Dado m, es computacionalmente intratable obtener un m´ tal que r(m) = r(m´).

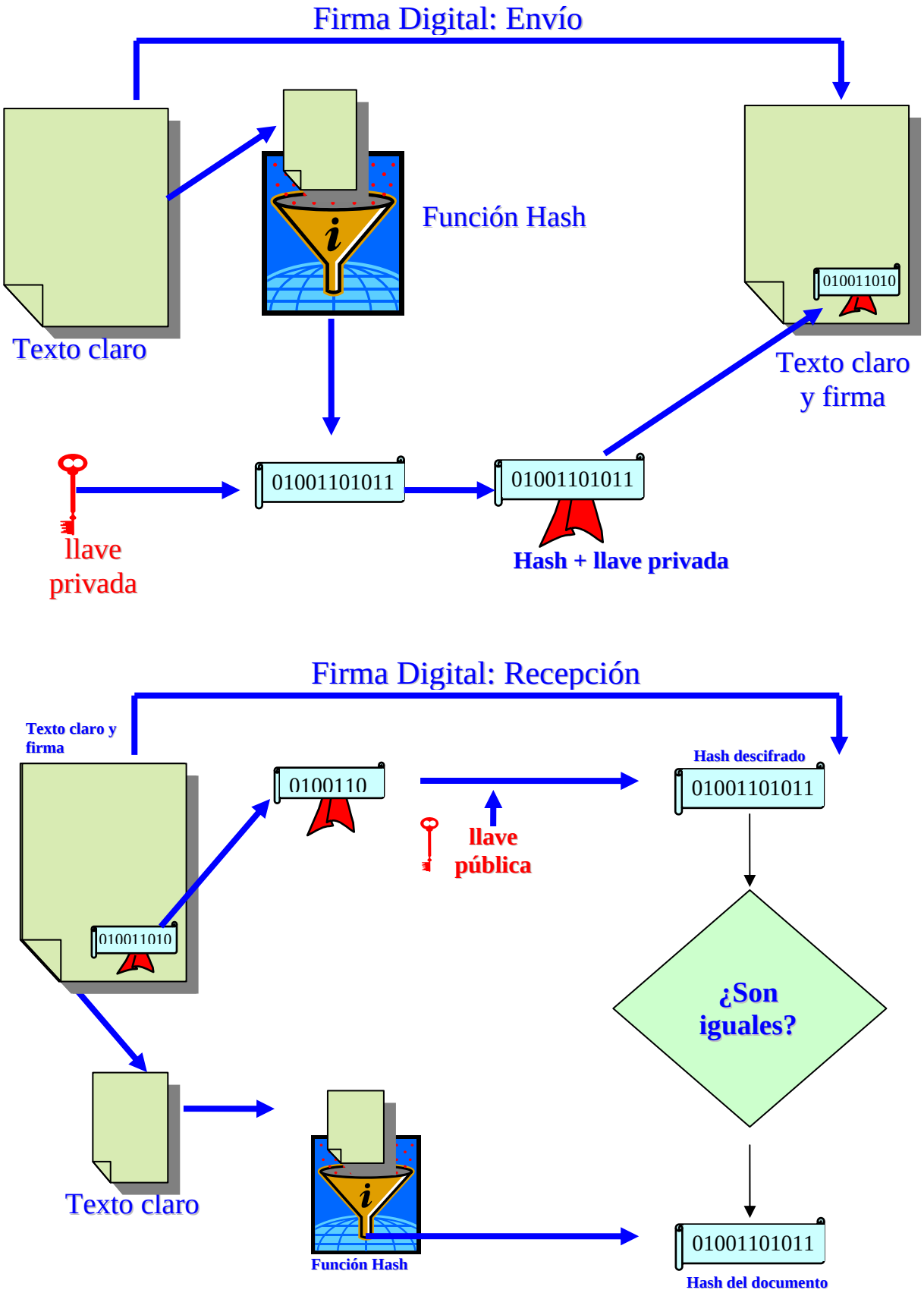


Figura 4.8.2.b Firma Digital aplicando funciones hash

La seguridad de la firma va siempre a depender de lo seguro de la función hash.

No existe ninguna forma de tomar la firma de alguien de un documento y ponerla en otro

No es posible alterar un mensaje firmado. Una alteración del documento firmado por más sencilla que esta sea será detectada en la verificación

4.8.3 Demostración de software utilizado para encriptar datos

4.8.3.1 Software: Criptografía Clásica

Autores: Cesar Andrade y David Cedillo

Descripción:

Este es un software didáctico muy sencillo que nos va a ayudar a explicar de manera muy rápida como se realizan los cifrados clásicos, tales como el Cifrado Cesar, Cifrado por Transformación Afín y el Cifrador Poli alfabético de Vigenère.

Este software está compuesto de cuatro formularios, los cuales vamos a explicar a continuación.

Entorno: Windows.

Instalación: Copie la carpeta cripto_clasico en la raíz **c:\.**

Menú Principal

Este Menú Principal está compuesto de dos opciones principales que son Cifrado Monoalfabético y Cifrado Polialfabético.

Estas opciones a su vez contienen otras subopciones que nos van a llevar a los formularios en donde vamos hacer pruebas de cifrado, así tenemos:

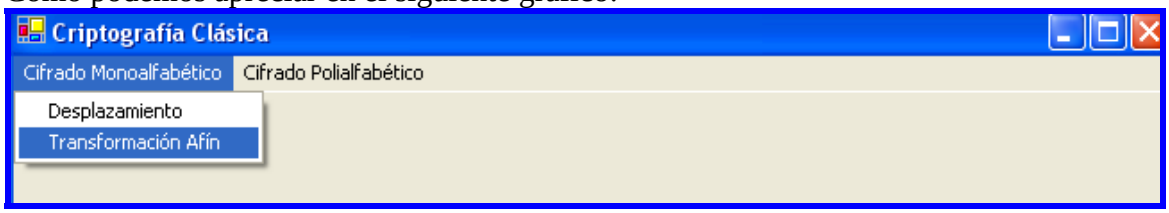
Cifrado Monoalfabético:

Mi + b mod 27
Afín

Cifrado Polialfabético:

Vigenère

Como podemos apreciar en el siguiente gráfico:



Menú Principal

Formulario Desplazamiento

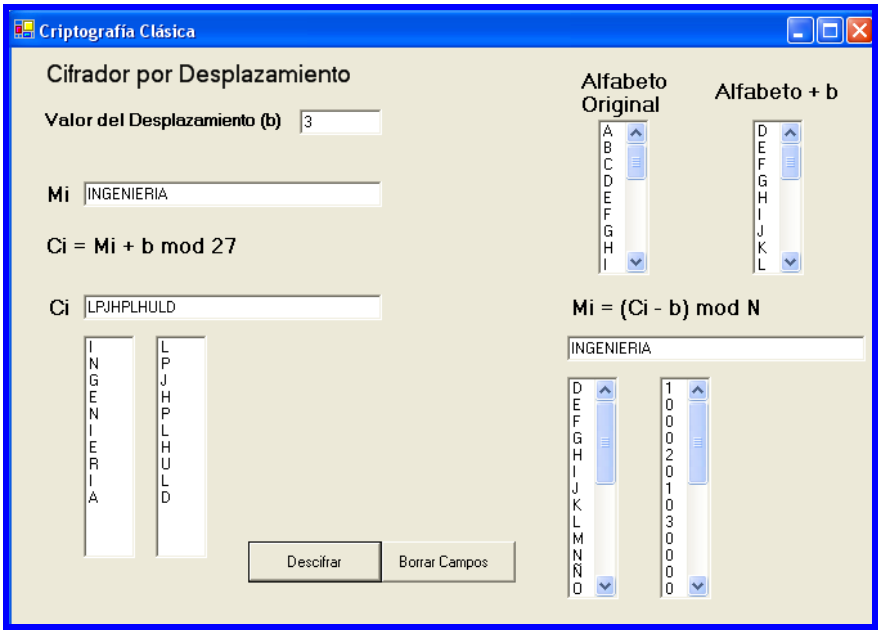
En este formulario podemos realizar prácticas de cifradores por desplazamiento, eéste tipo de cifradores aplica un desplazamiento de **b** espacios dentro de un alfabeto sobre el texto en claro.

La ecuación de cifra es: $\mathbf{C_i = (M_i + b) \bmod N}$

Donde **N=27**, el cuál es el número de elementos del alfabeto (A,B……Z)para las pruebas de este formulario.

La ecuación de descifrado es: $\mathbf{M_i = (C_i - b) \bmod N}$

El famoso cifrador del César usaba una constante **b = 3.**



Cifrador por Desplazamiento

Formulario Transformación Afín

En este formulario podemos realizar prácticas de cifradores por transformación afín, en este tipo de cifradores la operación de sustitución se realiza mediante la ecuación:

$$C_i=(a \times M_i + b) \bmod N$$

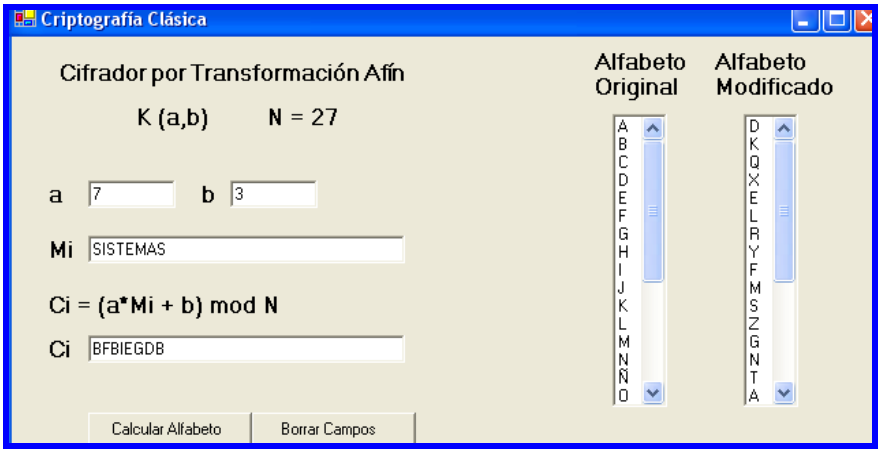
El descifrado usa la ecuación:

$$M_i = (C_i - b) \times a^{-1} \bmod N$$

Donde **a** es la constante de decimación, **b** la constante de desplazamiento y **N=27** el número de elementos del alfabeto.

En el siguiente ejemplo vamos a observar el cifrado con la clave:

K(a, b)= K(7, 3)



Cifrador por Transformación Afín

Formulario Cifrador de Vigenère

En este formulario podemos realizar prácticas del Cifrador de Vigenère, este cifrador aplica un desplazamiento de K espacios a los caracteres del texto en claro según el valor numérico asociado a cada uno de los caracteres de una clave **K** que se escribe cíclicamente bajo el mensaje.

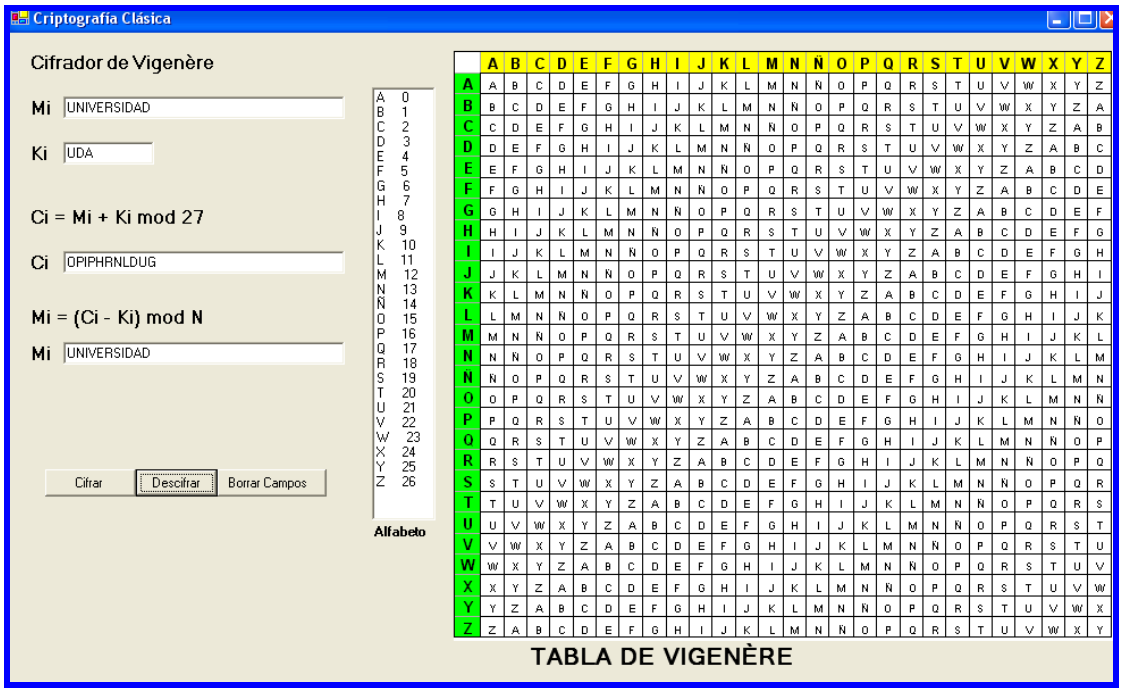
La longitud de **K** determinará el valor del periodo de cifra.

La transformación de cifra es:

$$C_i = (M_i + K_i) \bmod N$$

El descifrado será:

$$M_i = (C_i - K_i) \bmod N$$



Cifrador de Vigenère

4.8.3.2 Software: safeDES: Software de Ataque a la Fortaleza del EstándarDES

Autor: Ángel Jiménez Muñoz

Tutor: D. Jorge Ramió Aguirre

Este es un software correspondiente al trabajo fin de carrera de D. Miguel Ángel Jiménez Muñoz, que ha sido tutorizado y dirigido por D. Jorge Ramió Aguirre, profesor del Departamento de Lenguajes, Proyectos y Sistemas Informáticos (L.P.S.I.) de la Escuela Universitaria de Informática de la Universidad Politécnica de Madrid, España.

Descripción:

Cifrado, descifrado y ataques por fuerza bruta de forma similar a los desarrollados por RSA Challenge al algoritmo Data Encryption Standard DES. Operaciones con archivos o bien texto claro por teclado. Los textos y las claves pueden introducirse en formato ANSI o hexadecimal. Si bien el algoritmo implementa sólo el modo Libro Electrónico de Códigos ECB, es posible comprobar los ataques de los que fue objeto en modo CBC con el simple uso la calculadora científica de Windows en versiones 2000 o superiores. La aplicación cuenta una ayuda que permite el mejor seguimiento de las prácticas con este software.

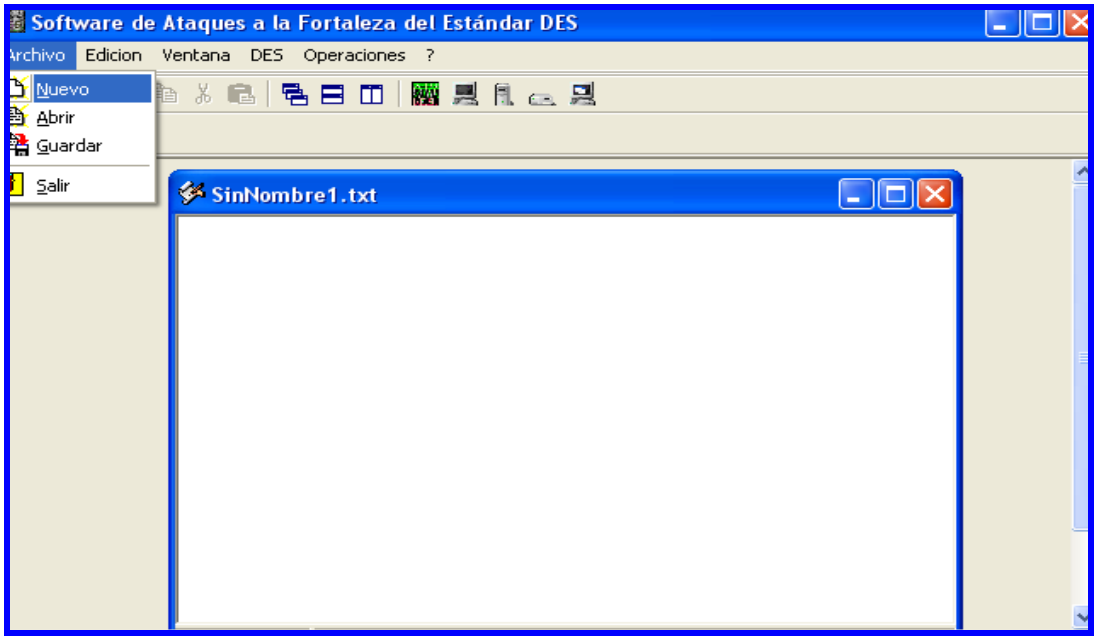
Entorno: Windows.

Instalación: Copie el archivo safeDES.zip en una carpeta de instalación. Al descomprimirlo, en dicha carpeta encontrará el archivo setup.exe. Ejecute este archivo e instale el programa en la carpeta C:\Criptolab\safeDES. Una vez instalado el programa borre los archivos de la carpeta de instalación.

NOTA: Este software es de dominio público y está prohibida su comercialización. Se puede obtener en la siguiente dirección electrónica http://www.criptored.upm.es

Menú Principal

Dentro del Menú Principal tenemos varias opciones, de las cuales las más importantes para nuestras prácticas describiremos a continuación.



Menú Principal

Archivo – Nuevo

En esta opción podemos ingresar un nuevo archivo de texto con la información que deseamos cifrar, como se indica en la figura anterior.

DES – Cifrar/Descifrar

Esta opción utilizamos para poder cifrar o descifrar información.

En el gráfico que vemos a continuación es donde ingresamos todos los requisitos que son necesarios ya sea para cifrar el archivo ingresado anteriormente o cifrar información ingresada por teclado. Además aquí debemos ingresar la clave con la que vamos a cifrar. Esta opción de cifrado nos dará como resultado un archivo con el nombre que nosotros deseemos, pero con extensión .cif.

Para descifrar un archivo debemos colocar la ubicación del archivo cifrado **(.cif)**, además de ingresar la clave de cifrado correspondiente. Esta opción de descifrado nos dará como resultado un archivo con el nombre que nosotros deseemos, pero con extensión **(.txt)**.

Opción DES

Luego de completar los requisitos como nombres de archivos, claves etc, debemos ir a la opción **Operaciones** y elegir el **Tipo de Operación** (Cifrar/Descifrar), y por último elegir **Comenzar**.

Opción Operaciones

Información más detallada acerca de este programa se encuentra en la ayuda del mismo.

4.8.3.3 Software: RSA-2000

Descripción:

El programa **RSA-2000 Versión 1.0**, es un programa diseñado para encriptar y decriptar ficheros de texto sin formato.

El algoritmo utilizado para la encriptación y decriptación es el RSA, desarrollado en 1977 por Ron Rivest, Adi Shamir y Leonard Adlerman.

Requisitos del sistema:

Este programa esta diseñado para trabajar en entorno MS-DOS. Debido a su presentación en CD, se deberán tener instalados los drivers del lector de CD bajo este entorno de trabajo, para poder ejecutar el programa.

El programa incluye un driver para el ratón, que se podrá instalar cada vez que se ejecute el programa si el usuario así lo desea.

Puesto que el entorno de trabajo es MS-DOS, se deberá tener en cuenta, que la ejecución desde una ventana MS-DOS de Windows, puede dar lugar a algunos problemas en el correcto funcionamiento del programa. Los problemas básicos que se han encontrado son:

No se deberá instalar el driver para el ratón, ya que si se instala, el ratón no funcionará dentro del programa debido a un conflicto con el driver de Windows.

Los contadores de tiempo que incluye el **RSA-2000**, pueden no funcionar correctamente.

Software adicional.

En el CD se incluye el programa **primos.exe,** el cual sirve para el cálculo de números primos, este programa esta ubicado en **\RSA-2000\PRIMOS**.

Cuando se ejecute **primos.exe**, aparecerá un mensaje para la introducción del nombre del fichero donde se almacenará una lista de números primos. Este archivo, se creará en **C:\RSA-2000\PRIMOS**. Después de introducir el nombre del archivo destino, el programa comenzará a calcular números primos e introducirlos en el anterior fichero, hasta que el usuario pulse una tecla. Cuando se termina la ejecución del programa, este creará un fichero en **C:\RSA-2000\PRIMOS**, llamado **tiempo.txt**, en el cual figurarán los siguientes datos referentes a la última ejecución del programa:

Número de primos generados.

Tiempo de cálculo empleado.

Este fichero tiene este nombre por defecto, por lo que sucesivas ejecuciones del programa **primos.exe** sobrescribirán dicho fichero, y, por tanto, almacenará los datos correspondientes a la última lista generada.

En el CD se incluye un fichero de números primos llamado **primos.dat**, ubicado en **\RSA-2000\PRIMOS** y generado con **primos.exe**. Este fichero contiene 15.541.554 números primos comprendidos entre el 2 y el 286.143.773

El programa **primos.exe**, puede ser utilizado por el usuario para la creación de nuevas listas de números primos, aunque se advierte que la creación de una nueva lista de igual o mayor tamaño que **primos.dat**, puede llevar un tiempo de cálculo considerable. Sirva como precedente que la generación del fichero **primos.dat**, supuso un tiempo de cálculo de: 61 horas, 30 minutos y 29 segundos.

Menús del RSA-2000.

En la parte superior de la pantalla, aparece la barra de menús. Los menús que presenta el **RSA-2000** son los siguientes:

Menú Principal

Ayuda

Si escogemos esta opción, aparecerá una pantalla con fondo azul, correspondiente al editor de textos del **RSA-2000**, en la que se visualizará el fichero de ayuda. Este fichero de ayuda esta ubicado en **\TEXTOS\Ayuda.txt**, dentro del CD.

Ayuda RSA-2000

Dentro del editor de textos, el ratón no esta disponible, y solo se podrán utilizar las teclas arriba, abajo, Avpag y Repag para desplazarnos por el fichero. Para salir del editor de textos, se deberá pulsar la tecla ESC.

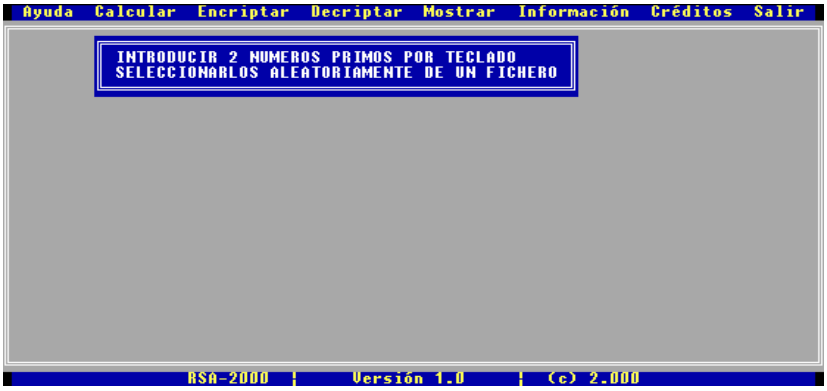
Calcular

En este menú, se calcula la clave pública de encriptado (E,N) y la clave privada de decriptado (D,N). Para calcular estas claves, se parte de dos números primos P y Q.

Aparecerá un submenú con dos opciones para introducir los dos números primos:

Introducir dos números primos por teclado.

Seleccionarlos aleatoriamente de un fichero.



Opción Calcular

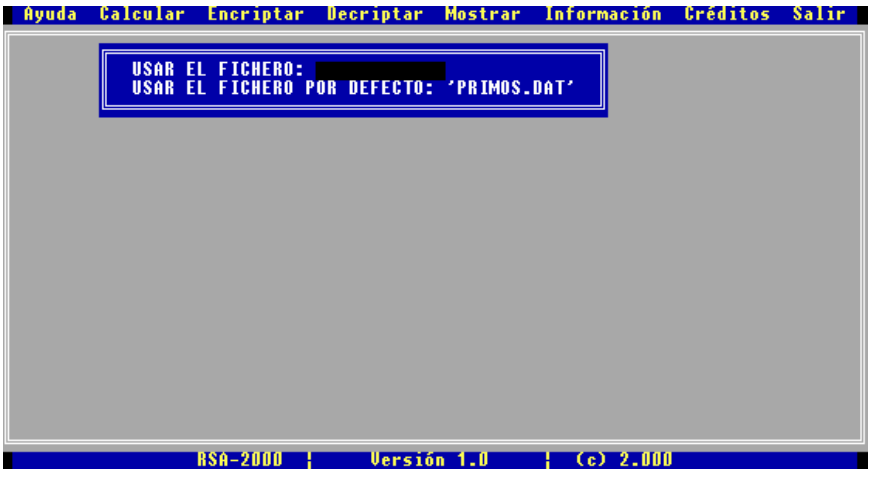
Si escogemos la primera opción, aparecerá otro submenú, en el que se nos pedirá primeramente él número P y luego él número Q, estos números no podrán tener más de 16 dígitos.

Ingresar por Teclado

Si escogemos la segunda opción, aparecerá otro submenú con dos opciones:

Usar el fichero: ----------

Usar el fichero por defecto “Primos.dat”.

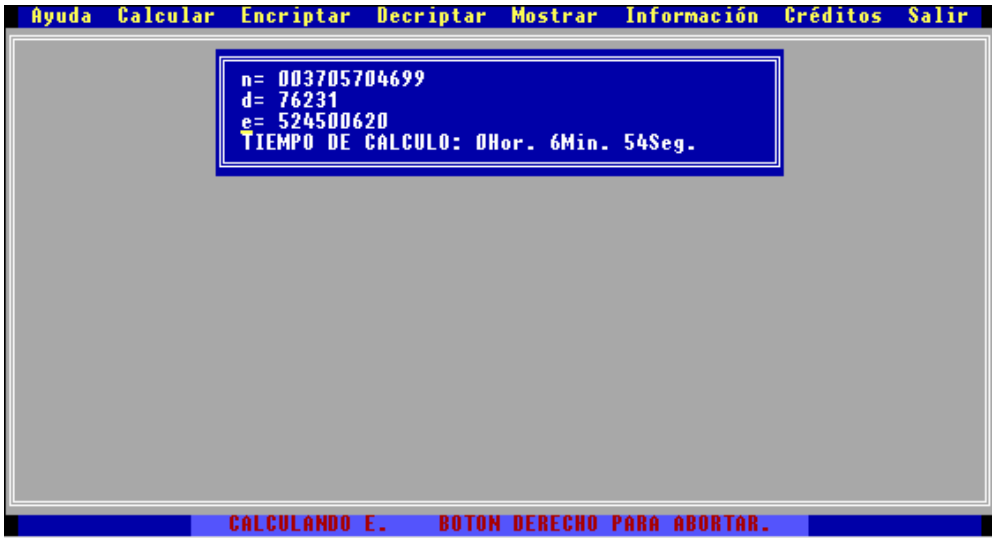


Utilizar ficheros

Si escogemos la primera opción, deberemos introducir el nombre de un fichero. Este fichero debe estar ubicado en **C:\RSA-2000\PRIMOS**, y debe contener al menos dos números primos. Los dos números primos se escogerán aleatoriamente del fichero.

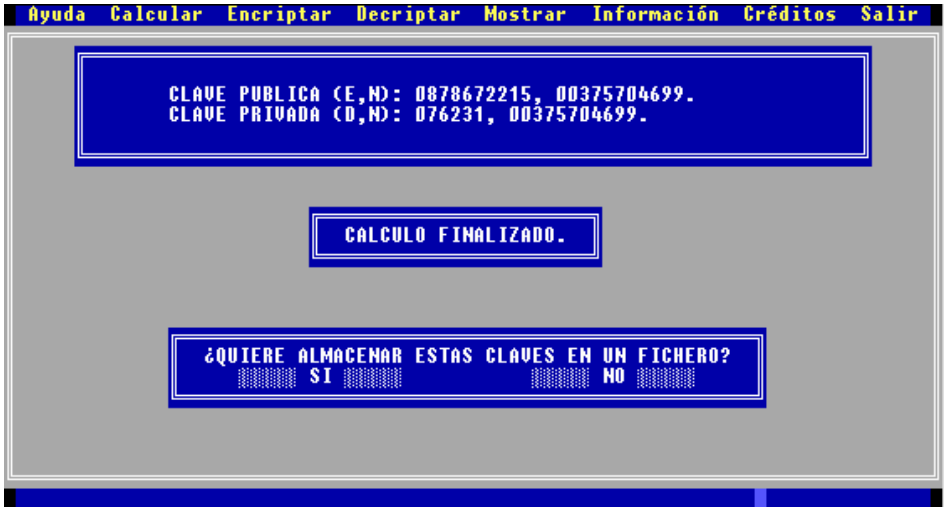
Si escogemos la segunda opción, utilizaremos para la selección de los dos números primos el fichero por defecto Primos.dat, que esta ubicado en **\PRIMOS**.

Una vez introducidos los dos números primos o el fichero del que se obtendrán, comenzará el proceso de cálculo, mostrándonos una ventana, en la que aparece él número n (n=p * q) perteneciente a las claves, el número d calculado, los sucesivos valores que se van probando para el número e, así como el tiempo de cálculo. El proceso de cálculo puede ser interrumpido pulsando el botón derecho del ratón.



Cálculo

Una vez finalizados los cálculos, se mostrará en una ventana la clave pública calculada, formada por los números e y n, y la clave privada, formada por los números d y n. A continuación se nos dará la opción de guardar las claves en un fichero. Si aceptamos, deberemos introducir el nombre del fichero en el que guardaremos la clave privada y el nombre del fichero en el que guardaremos la clave pública. Estos ficheros serán creados en **C:\RSA-2000\CLAVES**. A título informativo, en el directorio **C:\RSA-2000\ARCHIVOS** se creará el fichero **datos.txt**, que contendrá los valores: p, q, n y z obtenidos en la última ejecución de los cálculos.



Llaves Públicas y Privadas

Encriptar.

En primer lugar, se nos pedirá el nombre del archivo a encriptar. Este debe estar ubicado en el directorio **C:\RSA-2000\FUENTE** y debe ser un archivo de texto sin formato.

Fichero a encriptar

A continuación, aparecerá un submenú con dos opciones para introducir la clave de encriptado (clave pública):

Por teclado

Por fichero.

Introducir Claves

Si escogemos la primera opción, aparecerá otro submenú, en el que se nos pedirá primeramente él número E y luego el número N, estos números no podrán tener más de 16 dígitos.

Introducir Claves por teclado

Si escogemos la segunda opción, se nos pedirá el nombre del fichero en el que esta la clave pública de encriptado (formada por los números E y N). Este fichero debe estar ubicado en **C:\RSA-2000\CLAVES**. Si el fichero no se puede abrir o no existe, saldrá un mensaje de error y se saldrá del menú actual.

Una vez introducida la clave, comenzara el proceso de encriptado, mostrándonos un cuadro, con el porcentaje de archivo que ha sido encriptado.

Introducir Claves por fichero

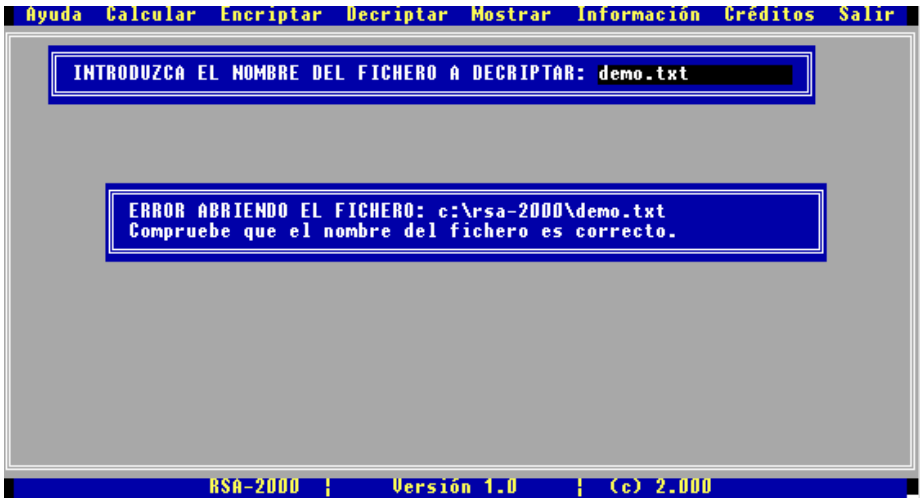
El archivo encriptado, se crea en **C:\RSA-2000** con el mismo nombre que el archivo fuente, pero con extensión .cod.

Proceso de encriptado

En el proceso de encriptado, se crea un archivo llamado **decimal.txt,** y ubicado en **C:\RSA-2000\ARCHIVOS**, el cual es necesario para la encriptación. Una vez terminada la encriptación, este archivo puede ser borrado.

Decriptar.

Al igual que en **Encriptar**, en primer lugar se nos pedirá el nombre del fichero a decriptar. Este debe estar ubicado en el directorio **C:\RSA-2000**, y debe tener extensión .cod. Además, debe haber sido encriptado con este programa. Si el fichero no es correcto, no existe o no se puede abrir, aparecerá un mensaje de error y se saldrá del menú actual.



Fichero a decriptar

A continuación, aparecerá un menú con dos opciones para introducir la clave de decriptado (clave privada):

Por teclado

Por fichero.

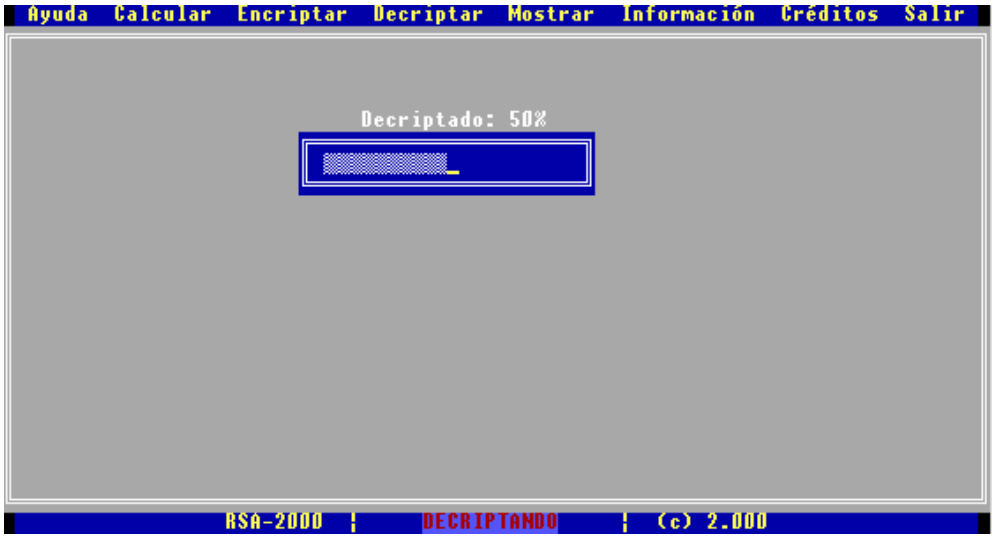
Introducir Claves de decriptado

Si escogemos la primera opción, aparecerá otro submenú, en el que se nos pedirá primeramente el número D y luego el número N, estos números no podrán tener más de 16 dígitos.

Introducir Claves por teclado

Si escogemos la segunda opción, se nos pedirá el nombre del fichero en el que esta la clave privada (formada por los números D y N). Este fichero debe estar ubicado en **C:\RSA-2000\CLAVES**. Si el fichero no se puede abrir o no existe, saldrá un mensaje de error y se saldrá del menú actual.

Una vez introducida la clave, comienza el proceso de decriptado, mostrándonos un cuadro, con el porcentaje de archivo que ha sido decriptado.



Proceso de decriptado

El archivo decriptado, se crea en **C:\RSA-2000**, con el mismo nombre que el archivo codificado pero con extensión .dec.

Fichero decriptado

En el proceso de decriptado, se crea un archivo llamado **caracter.txt** y ubicado en **C:\RSA-2000\ARCHIVOS**, necesario para el decriptación. Una vez terminada esta, dicho fichero puede ser borrado.

Mostrar.

Este menú esta dedicado a la visualización de diferentes archivos con los que puede trabajar el **RSA-2000**. Aparecerá un submenú con las siguientes opciones:

Ficheros de claves.

Ficheros a encriptar.

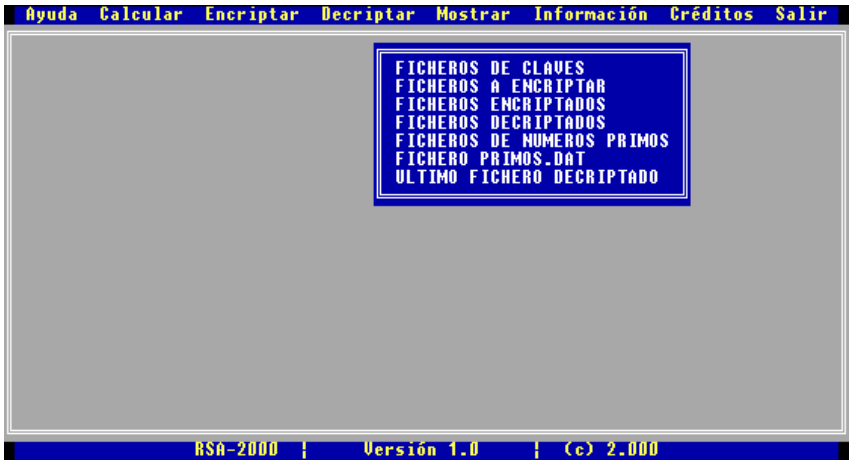
Ficheros encriptados.

Ficheros decriptados.

Ficheros de números primos.

Fichero Primos.dat.

Último fichero decriptado.



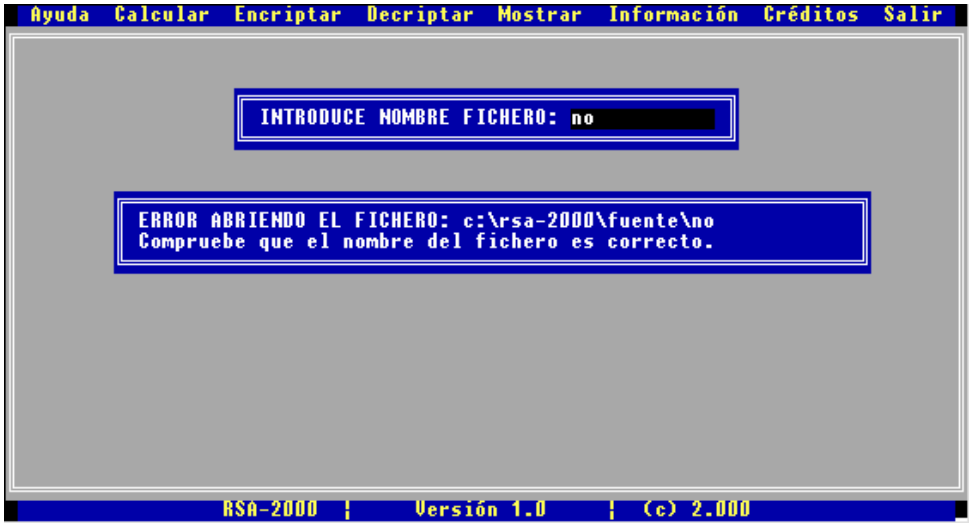
Opción Mostrar

Si la opción escogida es "Fichero Primos.dat", se editará directamente este archivo.

Si la opción escogida es "Último fichero decriptado", pueden ocurrir dos cosas. Si no se ha decriptado ningún fichero en la sesión actual, se visualizará un mensaje indicándonoslo. En caso contrario, se editará el último fichero decriptado.

Último fichero decriptado

Para el resto de opciones, se pedirá el nombre del archivo a mostrar (solo se pide el nombre, ya que los archivos deben estar ubicados en los directorios correspondientes al tipo de archivo, como ya se ha comentado). Si no se puede abrir el fichero o este no existe, se visualizará un mensaje de error, y se saldrá del menú actual.



Fichero a mostrar

Información.

Este menú esta dedicado a la visualización de un trabajo sobre Criptografía. Aparecerá un submenú, con las siguientes opciones:

Introducción histórica.

Introducción a la Criptografía.

Criptografía de clave secreta.

Criptografía de clave publica.

Sistema RSA.

Aplicaciones Criptográficas.

Bibliografía.

Opción Información

Después de escoger una opción, aparecerá la ventana del editor de texto del **RSA-2000**, en la que se visualizara el fichero de texto correspondiente a la opción escogida. Estos ficheros están ubicados en **\TEXTOS*.txt**, dentro del CD. Los ficheros son los siguientes: Historia.txt, Intro.txt, Secreta.txt, Publica.txt, Rsa.txt, Aplica.txt y Biblio.txt.

5: Conclusiones y Recomendaciones

La criptografía es un campo de mucha utilidad hoy en día, en una época en donde la información es poder, esto amerita su estudio, comprensión y su investigación por parte de las personas que están envueltas en la docencia y grupos de investigación de las diversas universidades de nuestro país.

Hemos llegado a la conclusión de la criptografía es un tema muy amplio, que no deja de ser ameno e interesante para su estudio, en la actualidad se vuelve cada vez más imperativo su conocimiento de parte de la gente relacionada con la informática, dada las aplicaciones que este campo tiene en las comunicaciones, a la que en nuestro medio está abriéndose camino.

Como recomendación podemos decir que el nivel de protección que queramos dar a nuestra información va a estar relacionada directamente con el valor de ésta, es decir el costo de mantener a salvo nuestra información nunca deberá superar el valor de la información que deseamos proteger.

En las universidades se debería abordar este tema como un complemento del aprendizaje de herramientas orientadas al Internet, dada su aplicación en el futuro cercano donde se prevé la creciente demanda en el campo del e-commerce, este estudio es fundamental para tener en cuento a seguridades necesarias para el desarrollo de transacciones seguras.

Otra de las muchas áreas en las que se debería tener en cuenta estas seguridades sería para el desarrollo de aplicaciones orientadas al campo del teletrabajo, ya que éste ha llegado a ser un factor indispensable.

6: Bibliografía.

-Introducción a la Criptografía.
Autor: Pino Caballero Gil.
Editorial RA-MA.

- Criptografía y Seguridad en Computadores
Manuel José Lucena López

- Libro electrónico de seguridad informática y criptografía
Autor: Jorge Ramió Aguirre
Universidad Politécnica de Madrid España
http://www.criptored.upm.es/

- Material de Apoyo Curso de Especialización de Telecomunicaciones
Autores: Ing. Marcelo Utard/Ing. Pablo Ronco
UDA-FIUBA ABRIL-JUNIO 2004

- http://rinconquevedo.iespana.es/rinconquevedo/Criptografia/criptografia.htm

- http://es.tldp.org/Manuales-LuCAS/doc-unixsec/unixsec-html/node304.html