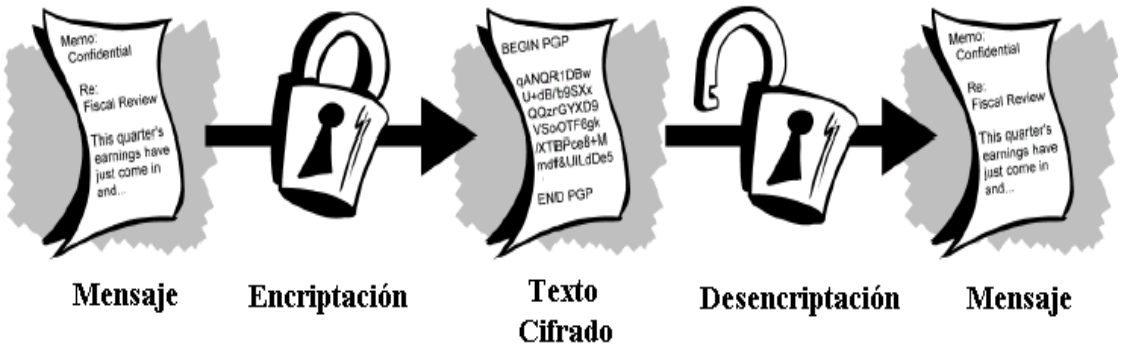
C A P Í T U L O II

C R I P T O L O G Í A



2.1 DEFINICIÓN

Criptografía, proviene del griego *krypto* que significa *oculto* y *graphein* que significa *escribir*. La *criptografía* es la ciencia de usar las matemáticas para cifrar y descifrar datos. Para nuestros fines, *cifrar* o *encriptar* es el proceso de transformar un texto conocido (*texto llano*) en una forma tremendamente complicada (*texto cifrado*), con el fin de que resulte ilegible para cualquiera que no posea la clave secreta. El proceso inverso de reconstruir el texto original partiendo del texto cifrado y de una clave de descifrado es el *descifrado* o *desencriptación***.**

El proceso de cifrado está basado en un algoritmo matemático y en una clave, que dado el algoritmo y el texto llano, nos genera el texto cifrado; puede verse como la cerradura y la llave, respectivamente. Veamos un ejemplo. Uno de los métodos de cifrado más antiguos es el llamado *cifrado de César*, que consiste en desplazar todas las letras del alfabeto en una cantidad conocida. Si el desplazamiento es, digamos cinco, eso significa que la letra A se transforma en la F (la letra que hay cinco posiciones a la derecha de la A en el alfabeto), la B en la G, la C en la H, y así sucesivamente. Es decir, convertimos cada letra de texto llano (primera fila) en una letra de texto cifrado (segunda fila) de acuerdo con la siguiente secuencia:

ABCDEFGHIJKLMNOPQRSTUVXYZ *(texto llano)*
FGHIJKLMNOPQRSTUVXYZABCDE *(texto cifrado)*

donde hemos supuesto que tras la Z viene la A. De este modo, JULIOCESAR se convierte en OAQNTHJYFX. Podemos denotar la clave como cinco (número de posiciones a desplazar); el algoritmo sería simplemente la sustitución de una letra por la que hay x posiciones a la derecha (x es la clave).

Para descifrar, no hay más que tomar el texto cifrado y sustituir cada letra por la que hay cinco posiciones a su izquierda (la O se convierte en la J, la A en la U...). Si usamos dígitos binarios en lugar de letras, podremos utilizar el sistema de César para cifrar todo tipo de archivos informáticos como: documentos de texto, archivos de audio, video, imagen e incluso archivos ejecutables.

El sistema Cesariano ya no se utiliza debido a la sencillez con la que se puede romper, o más correctamente, *criptoanalizar,* que es la ciencia de obtener el texto llano a partir del texto cifrado sin conocer la clave, o bien de obtener la propia clave a través de texto llano o cifrado.

El método de cifrado de César no es seguro porque resulta fácil presa del criptoanálisis. Para empezar, es susceptible de un *ataque de fuerza bruta*, que consiste simplemente en probar todas las posibles claves. Igual que si probásemos todas las combinaciones de una caja fuerte. Utilizando el alfabeto apuntado anteriormente, solamente tenemos veinticinco posibles claves (naturalmente, desplazar una letra 26 posiciones da igual resultado que desplazarla uno solo), lo que nos permite probar todas las combinaciones de forma cómoda. El número de posibles claves se conoce como *espacio de claves*.

La criptografía en realidad no es solo una rama de las matemáticas sino una disciplina que puede reunir otras áreas de la ciencia, sin embargo, es en las matemáticas en donde la criptografía moderna encuentra los fundamentos más trascendentales. En general la criptografía es el uso de problemas de difícil solución a aplicaciones específicas. Por ejemplo un problema de difícil solución es encontrar los factores de un número que es producto de dos números primos[9]. Cómo podemos aplicar este problema de difícil solución a

[9] Número entero positivo, distinto de 0 y que únicamente se puede dividir por sí mismo y por la unidad.

un caso específico, en particular, al tema de la confidencialidad de la información digital. Esto es un asunto del campo de la criptografía.
El fundamento y los procedimientos de operación para efectivamente dar solución a un problema específico constituyen un *criptosistema.*

El *criptoanálisis* es la actividad que se encarga de estudiar las debilidades de un criptosistema y su objetivo es el de encontrar soluciones fáciles al reto implantado en el criptosistema. Ambas actividades, la criptografía y el criptoanálisis se conocen colectivamente como *Criptología* [LIB 001].

2.2 HISTORIA e IMPORTANCIA

La criptografía tiene una larga historia. Históricamente, cuatro grupos de personas han usado y contribuido el arte del cifrado: los militares, el cuerpo diplomático, los diaristas y los amantes. De estos, los militares han tenido el papel más importante y han allanado el camino [LIB 001].

La criptografía se ha usado en el pasado sobre todo con fines bélicos. A sus generales en el frente, Julio César les enviaba mensajes escritos y los hacía crípticos escribiendo en lugar de cada letra del mensaje, la que está tres letras más adelante, cíclicamente (a la Z le sigue la A), en el alfabeto. Así la palabra ATAQUEN la escribía DWDTXHQ, incomprensible para el mensajero o para cualquier enemigo que lo interceptara.

Durante la Segunda Guerra Mundial, cuando la flota alemana dominaba el Atlántico Norte, el alto comando se comunicaba con ella por radio usando mensajes encriptados por la famosa máquina *Enigma*, mientras antenas inglesas los interceptaban sin que nadie pudiera descifrarlos. Un equipo de

matemáticos británicos efectuaron una serie de ataques criptoanalíticos logrando quebrar los códigos, delatar la ubicación y planes de las naves enemigas y cooperar en el cambio de rumbo de la guerra.

Actualmente vivimos tiempos gloriosos para las Matemáticas. Pocas veces antes en la historia se habían visto usos tan tangibles de su poder y de la confianza que la ciencia y la tecnología depositan en el rigor y la fuerza del pensamiento matemático. Una de las aplicaciones de mayor interés y en actual estado de cambio y ebullición, es el desarrollo de esquemas criptográficos que usan los gobiernos, los ejércitos, el comercio, la banca y con intensidad creciente, la Internet.

Modernamente los esquemas criptográficos son más sofisticados y robustos, y se les usa por la necesidad de privacidad de vastos aspectos de la vida. Hay necesidad de privacidad de nuestros expedientes médicos, de las transacciones comerciales, de movimientos bancarios, de cartas de amor, etc. La única manera de proteger los datos de una manera segura es la *criptografía*. Es curioso que una técnica que se remonta al antiguo Egipto sea hoy aplicada como solución a uno de los importantes retos de la Era Digital. Actualmente se dispone de una avanzada tecnología en métodos matemáticos que codifican los bits de tal forma que resulta imposible interpretarlos sin las fórmulas esenciales que los crearon, convirtiendo la información en algo inviolable.

En general, las posibilidades de desarrollo que brindan las innovaciones tecnológicas en telecomunicaciones y gestión de la información son ilimitadas: desde la consulta de saldos y movimientos de la cuenta corriente, pasando por la realización de transferencias y otras operaciones a través de un teléfono móvil con la última tecnología WAP o la compra-venta de valores y acciones

con la colaboración de agentes financieros software a través de Internet a precios muy ventajosos.

2.3 FUNDAMENTOS

2.3.1 LOS NÚMEROS ALEATORIOS

Las sucesiones de números o bits aleatorias son sucesiones de números o bits seleccionados al azar de forma uniforme, es decir, todo número o bit tiene la misma probabilidad de ser escogido.

¿Para qué sirven los Números Aleatorios?

En la vida cotidiana se utilizan números aleatorios en situaciones tan dispares como pueden ser los juegos de azar, en el diseño de una animación por ordenador, localización de errores en chips, transmisión de datos desde un satélite, finanzas, etc.

En general cuando se requiere una impredecibilidad en unos determinados datos, se utilizan números aleatorios. Aquí es donde entra en juego la criptografía. Así en la telefonía móvil digital GSM se utilizan para la asignación de una clave aleatoria que sirve para autentificar al usuario o por ejemplo también se utilizan para dar cierta seguridad a la asignación inicial de números secretos a las tarjetas de crédito. Para muchos sistemas criptográficos, como pueden ser DES, RSA, DSA, la utilización de números aleatorios es vital para su seguridad.

¿Qué son los Números Pseudoaleatorios?

Son unos números generados por medio de una función (determinista, no aleatoria) y que aparentan ser aleatorios. Estos números pseudoaleatorios se generan a partir de un valor inicial aplicando iterativamente la función. La sucesión de números pseudoaleatorios es sometida a diversos tests para medir hasta qué punto se asemeja a una sucesión aleatoria.

¿Por qué hay que recurrir a los Números Pseudoaleatorios?

Fundamentalmente porque las sucesiones de números pseudoaleatorios son más rápidas de generar que las de números aleatorios. Los dos principales inconvenientes de las sucesiones de números pseudoaleatorios es que a partir de un mismo valor inicial se genera la misma sucesión y que, en general, la sucesión es periódica.

¿Cómo se generan las sucesiones pseudoaleatorias?

Uno de los primeros métodos conocidos es debido a John Von Neumann (1903-1957). Una versión de este método consiste en partir de un número inicial de 10 cifras, elevarlo al cuadrado y seleccionar como nuevo número aleatorio las cinco cifras centrales. Siendo cautos en la elección del valor inicial e iterando el proceso se obtiene una sucesión de números pseudoaleatorios.

En la actualidad existe gran cantidad de métodos para generar sucesiones pseudoaleatorias.

¿Por qué son tan importantes para la criptografía?

La generación de números pseudoaleatorios es necesaria en diversos sistemas criptográficos, como por ejemplo a la hora de generar la clave en los sistemas DES, RSA, DSA, etc. Para muchos sistemas criptográficos, la utilización de números pseudoaleatorios es vital para su seguridad. Esto es, fundamentalmente, porque la elección de las claves tiene que ser aleatoria para que así sean impredecibles para un oponente. También se utilizan estos números en los cálculos intermedios que realizan los sistemas criptográficos.

2.3.2 LOS NÚMEROS PRIMOS

Un número es primo cuando es entero positivo, distinto de 0 y que únicamente se puede dividir por sí mismo y por la unidad.

Ejemplo: *1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...*

Notas:

- El 1 se considera primo en muchos casos, aunque sólo tiene un divisor.
- El 2 también cumple las características de número primo y es el único número primo que es par.
- La serie de números primos es ilimitada, o sea, que por grande que sea un número primo, siempre hay otro número primo mayor.

Los números primos siempre han sido unos números muy discutidos. Algunos matemáticos han intentado estudiar el porqué del orden que siguen, sin llegar a conclusiones. La aparente ausencia de un principio establecido de organización en la distribución o sucesión de los números primos había traído

de cabeza a los matemáticos durante siglos y proporcionando gran parte de su atractivo a la teoría de los números.

Los antiguos griegos y después de ellos los grandes matemáticos de la ilustración europea, como Pierre de Fermat, Leonard Euler y Carl-Fridrich Gauss, habían descubierto una variedad de teoremas interesantes relacionadas con los números primos. Sin embargo, hasta mediados del siglo XIX, las verdades más fundamentales sobre ellos permanecieron fuera del alcance de los matemáticos. Las principales eran dos: su distribución, es decir, la cantidad de números primos menores que un entero dado ***n*** (Conforme ***n*** tienda al infinito, la cantidad de primos se aproxima a $n/ln(n)$.) y las pautas de su sucesión, la escurridiza fórmula mediante la cual, partiendo de un número primo dado ***Pn***, uno podía determinar el siguiente, Pn+1. A menudo (quizás infinitamente a menudo, según una hipótesis), los números primeros sólo están separados por dos enteros, en pares como 5 y 7, 11 y 13, 41 y 43 o 9857 y 9859.

Los criptosistemas de clave pública, como el de Diffie y Hellman, están basados en propiedades matemáticas de los números primos. El problema de la ***Factorización*** es justo el inverso a la multiplicación. Si para multiplicar partíamos de la existencia de dos números y tratábamos de hallar su producto (para lo cual existen algoritmos claramente definidos), en el caso de la factorización partimos de un número y tratamos de hallar sus factores primos. Por ejemplo, si nos dan el 342, deberíamos concluir que 342=2*3*3*19.
Desgraciadamente (o afortunadamente, según se mire) no se conoce ningún algoritmo capaz de hacer esto de forma rápida y eficiente cuando el número que queremos factorizar es un número primo grande[10].

[10] Se considera que un número es grande si tiene una longitud de al menos 512 bits (~ 155 dígitos).

NÚMEROS PSEUDOPRIMOS

Se llaman números pseudoprimos los números que parecen primos pero no lo son. La única forma certera de determinar si un número es primo es dividir por todos los anteriores (bueno, podríamos quitar algunos números que sabemos no son primos, los pares, los terminados en 5, etc.) y comprobar que el resto no es cero en ninguno de ellos. Cuando el número es muy grande, este proceso es inviable, incluso con potentes ordenadores, por eso se utilizan métodos probabilísticos. Si el número cumple la prueba y no es primo, se dice que ese número es pseudoprimo.

Por ejemplo, si decimos que todos los primos son de la forma 6n+5, y el número elegido es el 35, diremos que 35 es un pseudoprimo porque cumple la condición 6n+5 y no es primo.

2.3.3 ARITMÉTICA MODULAR

En este tema estudiaremos la aritmética modular, es decir, la aritmética de las clases de congruencias, la cual simplifica los problemas teórico-numéricos sustituyendo cada entero por el resto de dividirlo entre un entero positivo fijo ***n***. Esto produce el efecto de sustituir el conjunto infinito **Z** de números enteros por un conjunto $\mathbf{Z}_n$ que sólo contiene ***n*** elementos. Encontraremos que se pueden sumar, restar y multiplicar los elementos de $\mathbf{Z}_n$ (igual que en **Z**), aunque encontramos algunas dificultades en la división. De este modo, $\mathbf{Z}_n$ hereda mucha de las propiedades de **Z**, pero al tratarse de un conjunto finito es más fácil trabajar con ellos.

Muchos problemas en los que se requieren enteros muy grandes pueden simplificarse con una técnica denominada *aritmética modular*, en la que se utilizan *congruencias* en vez de ecuaciones. La idea básica es elegir un determinado entero *m* (dependiendo del problema), llamado *módulo* y sustituir cualquier entero por el resto de su división entre *m*. En general, los restos son pequeños y, por tanto, es fácil trabajar con ellos. Antes de entrar en la teoría general, veamos un ejemplo sencillo.

¿ Hay un reloj en tu dormitorio?

Si sumas 3 horas cuando el reloj marca las 10 horas, *Dónde terminas?*

Así que 10 + 3 = 1 en el dormitorio.

Esta aritmética del reloj no es un poco extraño?. A algunos matemáticos les gusta mucho. Ellos la llaman **aritmética modular**, y una ventaja es que no se tiene que contar más allá de 12, en este caso.

Así que en la aritmética modular (o aritmética del reloj) elegimos un entero mayor que 1 como módulo, llamémoslo *m*, y nos limitamos a trabajar con los números 0, 1, 2, ..., m - 1. De otra forma, sólo consideramos los restos posibles en la división por ***m***. Representaremos con el símbolo $\mathbf{Z}_m$ al conjunto de estos enteros módulo ***m***.

MÁXIMO COMÚN DIVISOR

Definición.- Si $d \mid a$ y $d \mid b$ decimos que *d* es un *divisor común* (o *factor común*) de *a* y *b*; por ejemplo, 1 es un divisor común a cualquier par de enteros *a* y *b*. Si *a* y *b* son no nulos, ninguno de sus divisores comunes puede ser mayor que máx ($|a|,|b|$). Este es el *máximo común divisor* de *a* y *b* que denotaremos por mcd (*a, b*); siendo el *único* entero *d* que satisface:

1. $d \mid a$ y $d \mid b$ (por ser *d* un divisor común).

2. Si $c \mid a$ y $c \mid b \Rightarrow c \mid d$ (pues d es el mayor de los divisores comunes de a y b).

Sin embargo, el caso $a = b = 0$ debe ser excluido; cualquier entero divide a 0 y es, por tanto, un divisor común de a y b, por lo que, en este caso, no existe un *máximo* común divisor. Esta definición puede obviamente extenderse al máximo común divisor de cualquier conjunto de enteros (no todos nulos).

Una forma de encontrar el máximo común divisor de a y b es simplemente construir las listas de todos los divisores de a y todos los de b para buscar el mayor entero que aparece en ambas. Evidentemente basta con buscar la lista de los divisores positivos: si, por ejemplo, $a = 12$ y $b = -18$, los divisores positivos de 12 son 1, 2, 3, 4, 6, 12 y los de -18 son 1, 2, 3, 6, 9,18 con lo que inmediatamente vemos que el máximo común divisor es 6. Este método resulta muy tedioso cuando a o b son grandes, pero afortunadamente existe un método más eficiente, para calcular el máximo común divisor, llamado *algoritmo de Euclides*. Este método se basa en la siguiente observación.

Dados dos enteros a y b se verifica que $\mathrm{mcd}(a, b) = \mathrm{mcd}(b, r)$ cualesquiera que sean los enteros q y r verificando que $a = bq + r$.

Algoritmo de Euclides

$P1$ *Leer* a *y* b

$P2$ $c \leftarrow a\,, \quad d \leftarrow b$

$P3$ $q \leftarrow \left\lfloor c/d \right\rfloor\,, \quad r \leftarrow c - dq$

$P4$ *Si* $r = 0$ *entonces el* $\mathrm{mcd}(a,b) = d$. *FIN*

$P5$ *Si no* $c \leftarrow d\,, \quad d \leftarrow r$

$P6$ *ir a* $P3$

CONGRUENCIAS

Definición 2.1: Sea *n* un entero positivo y sean *a* y *b* dos enteros cualesquiera. Se dice que *a* es ***congruente*** con *b* módulo *n*, y lo denotamos por $a \equiv b \pmod{n}$, si *a* y *b* dan el mismo resto cuando se dividen entre *n*. Para ser más preciso, utilizando la notación del algoritmo de la división para expresar:

$$\left.\begin{array}{ll} a = qn + r & con \quad 0 \le r < a \\ b = q'n + r' & con \quad 0 \le r' < b \end{array}\right\} \quad a \equiv b \pmod{n} \Leftrightarrow r = r'$$

donde *q*, *q'* son los cocientes y *r*, *r'* son los residuos.

Nuestro primer resultado viene a dar una definición alternativa de congruencia módulo *n*.

Teorema 2.1: Para cualquier entero dado $n \ge 1$ se tiene que si, y sólo si *n* divide a la diferencia *a* – *b* y se denota con $n \mid (a - b)$.

Por ejemplo:

1. $128 \equiv 446 \pmod{53}$

   Al dividir 446 por 53 obtenemos resto 22, y al dividir 128 por 53 también obtenemos resto 22. Luego 446 y 128 son congruentes módulo 53.

   También 53 divide a (446 -128) o lo que es lo mismo $53 \mid (446 - 128)$
2. $8 \equiv 17 \pmod{3}$ *porque* $3 \mid (8-17)$
3. $21 \equiv 1 \pmod{20}$ *porque* $20 \mid (21-1)$
4. $7 \equiv -1 \bmod 8$ *porque* $8 \mid (7-(-1))$

Teorema 2.2: Para cualquier entero fijo $m \ge 1$, la relación congruencia módulo *m* tiene las siguientes propiedades:

a*) Reflexiva:* $a \equiv a \pmod{m}$ para cualquier entero *a*

b) *Simétrica:* Si $a \equiv b \pmod{m} \Rightarrow b \equiv a \pmod{m}$

c) *Transitiva:* Si $a \equiv b \pmod{m}$ y $b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$

Teorema 2.3: Sean a, b, c enteros y m entero positivo.

Si $a \equiv b \pmod{m}$ entonces:

a) $a+x \equiv b+x \pmod{m}$ para todo entero x.

b) $ax \equiv bx \pmod{m}$ para todo entero x

Si $a \equiv b \pmod{m}$ y $c \equiv d \pmod{m}$, entonces:

a) $a+c \equiv b+d \pmod{m}$

b) $a-c \equiv b-c \pmod{m}$ ó $a-d \equiv b-d \pmod{m}$

c) $ac \equiv bd \pmod{m}$

d) $a^n \equiv b^n \pmod{m}$ para todo entero positivo n

Ejemplo: Encontrar el residuo de dividir 2^{30} entre 15.

El problema es equivalente a encontrar cuál de las quince clases residuales módulo 15, que contienen a 0, 1, 2, ..., 14 respectivamente, contiene a 2^{30}. Más claramente, se trata de hallar un número a entre 0 y 14 tal que cumpla que $2^{30} \equiv a \pmod{15}$, o lo que es lo mismo; encontrar un a que sea igual al residuo de dividir 2^{30} entre 15.

Hay que notar que: $2^4 \equiv 1 \pmod{15}$ porque $15 | (2^4-1)$. Entonces obtenemos: $(2^4)^7 \equiv 1^7 \pmod{15}$, esto es, $2^{28} \equiv 1 \pmod{15}$. Multiplicando a ambos lados 2^2 se tiene $(2^{28})_x 2^2 \equiv 1_x 2^2 \pmod{15}$, o sea, $2^{30} \equiv 4 \pmod{15}$.

Por tanto el residuo de dividir 2^{30} entre 15 es 4.

Definición 2.2: Si $n > 0$ no divide a $(b - c)$, se dice entonces que b y c son incongruentes módulo n y se escribe como $b \neq c \pmod{n}$.

Lemas:

1. Sea m un divisor de a, b y n y sean $a' = a/m$, $b' = b/m$ y $n' = n/m$. Entonces: $ax \equiv b \pmod{n} \quad \Leftrightarrow \quad a'x \equiv b' \pmod{n'}$
2. Sean a y n primos entre sí, m un divisor de a y b y sean $a' = a/m$ y $b' = b/m$. En este caso: $ax \equiv b \pmod{n} \quad \Leftrightarrow \quad a'x \equiv b' \pmod{n}$

Algoritmo de Resolución de Congruencias Lineales

P1 Dado la congruencia general de la forma $a\,x \equiv b \ (mod\ n)$, calculamos $d = \text{mcd}(a, n)$ aplicando el *Algoritmo de Euclides* y vemos si d divide a b.

- Si d no divide a b, FIN. La congruencia no admite solución.
- Si d divide a b la congruencia admite solución y vamos al Paso 2

La estrategia general es reducir $|\ a\ |$ hasta hacer $a = \pm 1$ ya que, en este caso, la solución particular es $x_0 = \pm b$.

P2 Como d es un divisor de a, b y n, implica que podemos reemplazar la congruencia original por $a'x \equiv b' \ (mod\ n')$, donde $a'=a/d$, $b'=b/d$ y $n'=n/d$. Por lo que a' y n' son primos entre sí.

P3 Podemos ahora dividir esta nueva congruencia entre $m = \text{mcd}(a', b')$ para obtener $a''x \equiv b'' \ (mod\ n')$, donde $a'' = a'/m$ es primo con $b'' = b'/m$ y con n'.

- Si $a'' = \pm 1$, $x_0 = \pm b''$ es la solución buscada.
- En caso contrario vamos al Paso 4.

P4 Observando que $b'' \equiv b'' \pm n' \equiv b'' \pm 2n' \equiv b'' \pm 3n' \equiv \ldots \ (mod\ n')$ somos capaces de reemplazar b'' por alguno de sus congruentes $b''' = b'' + kn'$ de tal forma que $\text{mcd}(a'', b''') > 1$; aplicando ahora el Paso 3 a la congruencia $a''x \equiv b''' \ (mod\ n')$ podemos reducir $|\ a''\ |$.

Ejemplo: Para comprender mejor el algoritmo anterior, vamos a ir aplicando cada paso de dicho algoritmo a la congruencia $10x \equiv 6 \pmod{14}$.

- ***P1:*** Mediante el Algoritmo de Euclides obtenemos que $d = \text{mcd}(10,14)$ con lo que $d=2$ el cual divide a 6 y, por tanto, existe solución.
- ***P2:*** Dividimos la congruencia original entre $d = 2$ para obtener $5x \equiv 3 \pmod{7}$.
- ***P3***: Al ser $m = \text{mcd}(5,3) = 1$, este paso no surte efecto.
- ***P4:*** Si tomamos k=1 y reemplazamos en la ecuación $b''' = b'' + kn'$ obtenemos $b''' = 3 + (1*7) = 10$, siendo 10 divisible por 5, por lo que podemos reemplazar la congruencia $5x \equiv 3 \pmod{7}$ por la congruencia $5x \equiv 10 \pmod{7}$.
- ***Volvemos a P3***

  Calculamos $m = \text{mcd}(5,10)=5$ y dividimos la congruencia lineal $5x \equiv 10 \pmod{7}$ por m (que es primo con 7), con lo que resulta la congruencia final $x \equiv 2 \pmod{7}$.

  Por tanto, $x_0=2$ es una solución particular de la congruencia.

2.3.4 TEST DE PRIMALIDAD

Existe un problema práctico en relación a la teoría que hemos considerado en este capítulo y es cómo determinar si un número entero es primo o compuesto (no primo). Un método sencillo de comprobar la primalidad de

un número es el test de las divisiones sucesivas el cual se basa en el siguiente lema:

- Un entero $n > 1$ es compuesto si, y sólo si, es divisible por algún primo $p \leq \sqrt{n}$.

Este método es efectivo para enteros pequeños, ya que no hay que considerar demasiados números primos, pero cuando *n* es grande se vuelve impracticable. En criptografía se utilizan con regularidad enteros con algunos cientos de dígitos decimales, este método requiere testar gran cantidad de números primos y hasta la más avanzada supercomputadora tardaría demasiado tiempo en realizar dicha tarea. Afortunadamente, existen otros algoritmos alternativos (utilizando algunas sofisticadas teorías de números), para testar la primalidad de números enteros grandes, más eficientemente. Algunos de éstos test rápidos son algoritmos probabilísticos, y uno de los mejores es el de *Miller y Rabin*, el cual detecta si un número entero es primo, pero puede declarar, incorrectamente, como primo un número compuesto; esto puede parecer un defecto desastroso, pero de hecho, la probabilidad de que esto ocurra es muy pequeña, tan pequeña como la probabilidad de un error computacional debido a un fallo de la máquina, por lo que, en la práctica, resulta ser muy seguro.

TEST DE PRIMALIDAD DE MILLER- RABIN

Es el algoritmo más empleado debido a su facilidad de implementación y se basa en escoger un número aleatorio ***a*** y efectuar una serie de operaciones entre ***a*** y ***p***, siendo ***p*** el número que deseamos saber si es primo o compuesto. Si se cumplen ciertas propiedades, sabremos con un grado de certeza determinado que ***p*** puede ser primo. Repitiendo este test muchas veces con

diferentes valores de ***a***, podemos aumentar nuestra confianza en la primalidad de ***p*** tanto como queramos.

A continuación se presenta los detalles del algoritmo:
Sea ***p*** el número que deseamos saber si es primo o compuesto. Se calcula ***b***, siendo ***b*** el número de veces que 2 divide a (p - 1), es decir, 2^b es la mayor potencia de 2 que divide a (p - 1). Calculamos ***m***, tal que $p = 1 + 2^b m$.

1. Escoger un número aleatorio a < p.
2. Sea j = 0 y z = a^m (mod p).
3. Si z = 1, o z = p - 1, entonces p pasa el test y puede ser primo.
4. Si j > 0 y z = 1, p no es primo.
5. Sea j = j + 1. Si j = b y z ≠ p - 1, p no es primo.
6. Si j < b y z ≠ p - 1, entonces $z = z^2$ (mod p). Volver al paso (4).
7. Si j < b y z = p - 1, entonces p pasa el test y puede ser primo.
8. p no es primo.

Si la prueba no falla para un valor determinado de ***a***, este se llama un **testigo**. Se puede mostrar que la probabilidad de que ***p*** sea compuesto es no mayor que $\frac{1}{4}$. Si repetimos con ***k*** testigos, esa probabilidad se reduce a $\frac{1}{4^K}$.

Se recomienda ejecutar el test como mínimo cinco veces para alcanzar un alto grado de certeza.

2.4 CLASIFICACIÓN

Hay dos tipos básicos de Criptosistemas: los simétricos (también conocidos como convencionales o de clave secreta) y los asimétricos (también conocidos como los de clave pública). A continuación se detallan cada uno de ellos.

CRIPTOGRAFÍA SIMÉTRICA O DE CLAVE PRIVADA

Los cifrados simétricos requieren que tanto el emisor como el receptor tengan la misma clave. Esta clave es usada por el emisor para cifrar los datos, y de nuevo por el receptor para descifrar los datos, como se muestra en la *figura 2.1*.

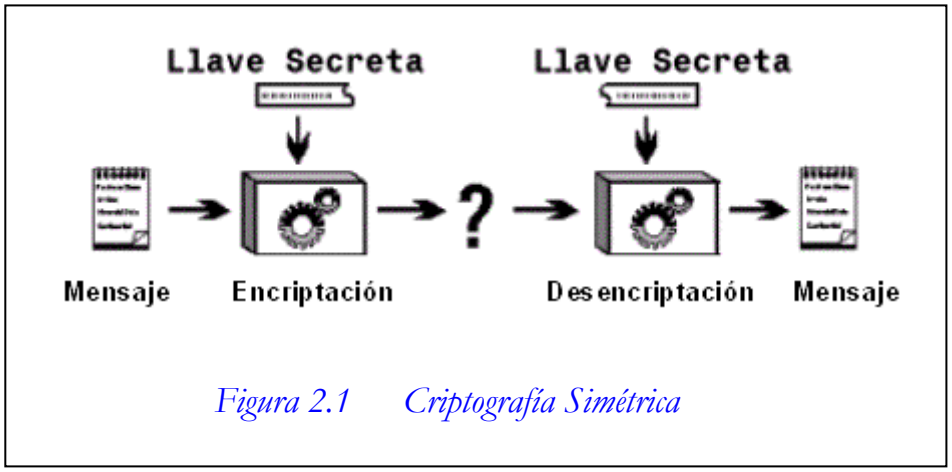


Figura 2.1 Criptografía Simétrica

Cuando la clave de cifrado es la misma que la de descifrado (y, por tanto, los algoritmos de cifrado y de descifrado coinciden) se habla de algoritmo de cifrado simétrico. Este es el tipo de cifrado que ha dominado la historia de la criptografía hasta hace un par de décadas.

Existen muchos algoritmos de clave simétrica resistentes al criptoanálisis: DES, RC4, IDEA, CAST, Triple-DES. Los algoritmos simétricos se clasifican en: *algoritmos de cifrado en bloque* [11] y *algoritmos de cifrado en flujo* [12].

Para cifrar, se coge el mensaje *M* y se le aplica la operación matemática *Ck* (es decir, se usa el algoritmo de cifrado *C* con la clave *k*), obteniendo *Ck(M)*. Para descifrar el mensaje, se aplica la operación inversa *Dk*. Puesto que *Ck* y *Dk* son operaciones inversas, se tiene Dk(Ck(M)) = M, esto es, el mensaje original. Esto requiere que el emisor y el receptor del mensaje utilicen tanto el mismo algoritmo como la misma clave.

La comunicación segura entre el emisor y el receptor pasa por que ambos, y nadie más, conozca la clave *k*. Pero, ¿cómo hacemos llegar la clave *k* de uno a otro? Podría enviarse por un canal seguro, esto es, uno que no pueda ser interceptado o espiado; pero ¿qué sentido tiene el cifrado si se dispone de un medio seguro de comunicación? El cifrado es para evitar a los fisgones; si no hay posibilidades de fisgar, no hay necesidad de cifrar.
Segundo problema: si la clave se filtra en cualquier momento a un tercero, la seguridad desaparece. Una clave comprometida [13] desbarata todo el sistema de comunicación segura basado en el cifrado.

Estos dos problemas: el intercambio y la gestión de las claves se resuelven con los sistemas asimétricos o de clave pública.

[11] Sistema criptográfico que cifra de bloque en bloque, usualmente cada bloque es de 128 bits.
[12] Sistema criptográfico que cifra de bit en bit.
[13] La clave puede estar en poder de un tercero no autorizado.

CRIPTOGRAFÍA ASIMÉTRICA O DE CLAVE PÚBLICA

Los cifrados asimétricos son mucho más flexibles desde el punto de vista de la administración de claves. Cada usuario tiene un par de claves: una clave pública y una clave privada. Los mensajes cifrados con una clave pública pueden ser descifrados solamente por la clave privada, ver *figura 2.2*. La clave pública puede ser ampliamente diseminada, mientras que la clave privada se mantiene en secreto.

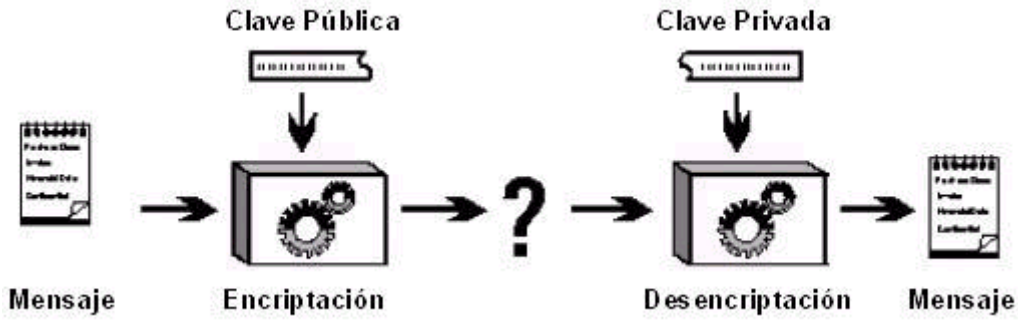


Figura 2.2 Criptografía Asimétrica

Supongamos que tanto Ana como Belén tienen un par de claves pública/privada. Si Ana desea enviar un mensaje a Belén, los pasos a seguir son los siguientes:

- Ana obtiene la clave pública de Belén *k*
- Ana compone un mensaje *M* y lo cifra con la clave pública de Belén.
- Ana envía el mensaje cifrado *Ck(M)*
- Belén recibe el mensaje y le aplica su clave privada *k'* obteniendo Dk'(Ck(M)) = M

Es decir, cualquiera puede cifrar un mensaje y enviárselo a Belén, pero solamente ella podrá descifrar los mensajes que le llegan, nadie más, ni siquiera Ana podrá descifrar el mensaje que acaba de cifrar.
Véase que este elegante esquema evita los peligros de enviar la clave por un conducto inseguro; de hecho la clave pública puede ser tan diseminada como un número de teléfono. Fausto va a tener ahora más difícil, ya que acceder a la clave pública le servirá para enviar mensajes, pero no para leer los que se envían Ana y Belén entre sí. Incluso si obtuviese la clave secreta de Ana, ello le permitiría leer los mensajes que Belén envía a Ana, pero no las respuestas de Ana a Belén. Por supuesto, permanece la obligación por parte de Ana y Belén de guardar celosamente sus respectivas claves privadas. Pero Fausto ya no puede aprovecharse del intercambio de claves secretas.

La criptografía de clave pública se asemeja a un buzón de correos. Cualquiera puede introducir una carta en el buzón, pero solamente el poseedor de la llave del buzón podrá abrirlo para acceder a su contenido.

Las claves pública y privada no son independientes, y en teoría es posible obtener la clave privada a partir de la clave pública. Pero en la práctica, el volumen de cálculos matemáticos que ha de realizarse es demasiado grande para que pueda llevarse a cabo. Como en los sistemas de clave simétrica, el "truco" es hacer que haya tantas claves posibles que no resulte práctico, aun cuando sea posible en teoría. En la actualidad, los sistemas de clave pública más utilizados son el RSA y el Diffie-Hellman.

El cifrado asimétrico también adolece de dos problemas. La verificación de las claves públicas es un paso muy importante. El hecho de no verificar que la clave pública pertenece realmente a Belén deja abierta la posibilidad de que

Ana esté usando una clave pública cuya clave privada asociada esté en manos del enemigo.

El segundo problema es de índole práctica, se refiere a la eficiencia. Los algoritmos de clave pública son lentos. El cifrado de un mensaje mediante cifrado de clave pública es del orden de mil veces más lento que mediante un algoritmo de clave simétrica. Los ordenadores son cada vez más rápidos, pero cuando hay grandes cantidades de información por cifrar o descifrar (pensemos, por ejemplo, en una base de datos protegida mediante cifrado) puede llegar a ser una verdadera dificultad.

Estos problemas, y algunos otros (por ejemplo, ciertos ataques criptoanalíticos) se evitan mediante un *sistema híbrido* que tome lo mejor de los dos mundos. ¿Queremos rapidez y brevedad? Pues creamos una clave simétrica y ciframos el mensaje con ella. ¿Queremos enviar de forma segura la clave simétrica al destinatario? Pues ciframos la clave simétrica con la clave pública del destinatario.

Supongamos que Ana quiere enviar un mensaje a Belén. Lo que hace es lo siguiente:

- Crea una clave simétrica *K* y cifra el mensaje con dicha clave. Sea Ck(M) el resultado.
- Cifra la clave simétrica con la clave pública de Belén. El resultado es Ckp(K)
- Envía a Belén dos cosas: el mensaje (cifrado con la clave simétrica *K*) y la clave simétrica *K* cifrada con la clave pública de Belén.

Cuando Belén recibe el "paquete", procede a la inversa:

- Toma Ckp(K) y lo descifra usando su clave privada. El resultado es *K*
- Usa *K* para descifrar el mensaje Ck(M). El resultado es *M*

Es decir, la clave pública se usa para cifrar; pero lo que se cifra no es el mensaje, sino la clave simétrica con que va cifrado el mensaje. De ese modo hacemos llegar al destinatario la clave *K*, y podemos hacerlo por medios inseguros de transmisión.

2.5 APLICACIONES

2.5.1 PGP (Pretty Good Privacy)

"Intimidad bastante buena", es un sistema de encriptación por llave pública escrito por Phil Zimmermann y consiste en un paquete completo de seguridad de correo electrónico que proporciona confidencialidad, validación de identificación, firmas digitales y compresión. Debido a su calidad, gratuito, fácil manejo, virtualmente inviolable, y fácil disponibilidad en las plataformas MSDOS, Windows, Unix, Macintosh y otros ordenadores, es de amplio uso hoy en día. PGP permite intercambiar ficheros y mensajes con intimidad, autenticación y comodidad. 'Intimidad' quiere decir que sólo podrán leer el mensaje aquellos a quienes va dirigido. 'Autenticación' quiere decir que los mensajes que parecen ser de alguien sólo pueden venir de esa persona en particular. 'Comodidad' quiere decir que la intimidad y la autenticación se consiguen sin los problemas de gestión de claves asociados a otros programas de criptografía convencional.

No se necesitan canales seguros para intercambiar claves entre usuarios, por lo que PGP resulta mucho más fácil de utilizar. Esto se debe a que PGP está basado en la potente criptografía de "clave pública". PGP combina la comodidad del criptosistema de clave pública de RSA con la velocidad de la criptografía convencional, con resúmenes de mensajes para firmas digitales, compresión de datos antes de encriptar, con un buen diseño ergonómico y con una completa gestión de claves.

2.5.2 PEM (Privacy Enhanced Mail)

"Correo con confidencialidad mejorada" nació a partir de las necesidades de transferir información criptográfica a través del correo electrónico, principalmente. PEM es un estándar oficial de Internet y se describe en cuatro RFC[14]: del RFC 1421 al RFC 1424. De manera general, el PEM cubre el mismo territorio que el PGP. Sin embargo, también tiene algunas diferencias respecto al PGP en cuanto a enfoque y tecnología [LIB 001]. Este protocolo fue el resultado del trabajo conjunto del Internet Engineering Task Force (IETF), PEM Working Group, el Internet Research Task Force (IRTF) y el Privacy and Security Research Group (PSRG).

PEM define procedimientos para proporcionar servicios de seguridad (confidencialidad, autenticación, integridad y no repudio de origen) en mensajes de texto (y sólo de texto) de correo electrónico mediante la utilización de un mecanismo de gestión de claves basado en certificados.

Lo habitual en PEM es utilizar algoritmos de clave secreta para el cifrado de los datos y algoritmos de clave pública para la administración de claves y la

gestión de firmas digitales. Aunque no es obligatorio el uso de algoritmos de clave pública en PEM, es recomendable hacerlo para aprovechar las ventajas que ofrecen este tipo de algoritmos en cuanto a la administración de claves se refiere, ya que, al contrario de lo que sucede con los algoritmos de clave privada, no es necesario un canal de comunicación seguro para efectuar el intercambio de claves secretas.

2.5.3 S/MIME (Secure / Multipurpose Internet Mail Extension)

Extensiones Seguras Multipropósito de Correo en Internet, desarrollada inicialmente por RSA Data Security Inc., proporciona una forma de transmitir datos MIME de forma segura: autenticación, integridad (mediante firmas digitales) y confidencialidad (usando encriptación).

Estos servicios se aplican en la capa de aplicación y son proporcionados mediante la utilización de criptografía extremo a extremo entre el emisor de un mensaje y su recipiente. Para suministrar el servicio de firma digital y la gestión de claves de encriptación se emplea criptografía asimétrica y para el servicio de encriptación se usa criptografía simétrica.

Aunque ha sido desarrollada para sistemas de mensajería, su aplicación no se limita a este tipo de entornos, sino que puede ser empleada en cualquier mecanismo de transporte que opere con datos MIME.

S/MIME es un protocolo para enviar correo seguro por la Internet y trabaja sobre MIME, el cual es un protocolo de intercambio de objetos a través de

[14] Request for Comment (petición de comentario).- Es una comunicación a través de una serie de documentos técnicos. Los RFC se numeran en secuencia.

Internet y es el formato estándar industrial para el correo electrónico que define como se estructura el cuerpo del mensaje, S/MIME agrega firmas digitales y encriptación a ese formato. Este formato estandarizado permite a usuarios de diferentes programas de correo electrónico comunicarse unos con otros. Para enviar y recibir correo electrónico seguro usando Identificación Digital, debe estar trabajando con programas que soporten S/MIME.

Algunas de las aplicaciones que soportan Correo Electrónico Seguro son Outlook Express, la aplicación e-mail incluida con MS Explorer 4.0, Netscape Messenger, Deming, Frontier , Pre-mail, Opensoft, Connectsoft y Eudora.

2.5.4 EFS (Encrypted File System)

El Sistema de Archivos Encriptado proporciona la tecnología principal de encriptación de archivos para almacenar archivos del sistema de archivos NTFS de Windows NT encriptados en disco. EFS pretende resolver las preocupaciones de seguridad que surgen en relación con herramientas disponibles en otros sistemas operativos que permiten a los usuarios acceder a archivos desde un volumen NTFS sin una verificación de acceso. Con EFS, los datos en archivos NTFS están encriptados en disco. La tecnología de encriptación utilizada está basada en clave pública y se ejecuta como un servicio integrado con el sistema lo que facilita su gestión, hace difícil los ataques y es transparente para el usuario.

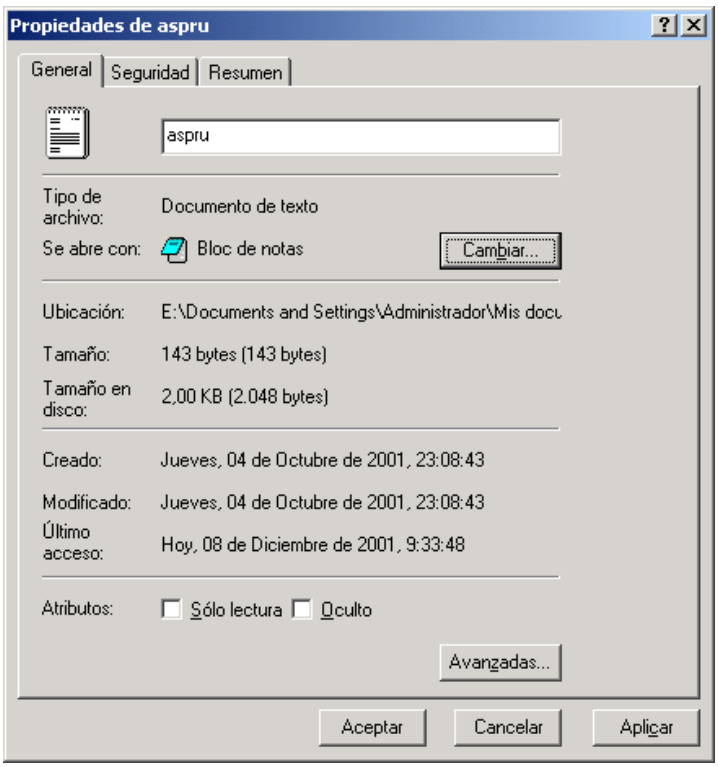
Si un usuario que intenta acceder a un archivo NTFS encriptado dispone de la clave privada de ese archivo, podrá abrirlo y trabajar con él transparentemente como un documento normal. Un usuario sin la clave privada del archivo tiene

denegado el acceso. Cuando cifre un volumen, tenga en cuenta que sufrirá cierta degradación en el rendimiento.

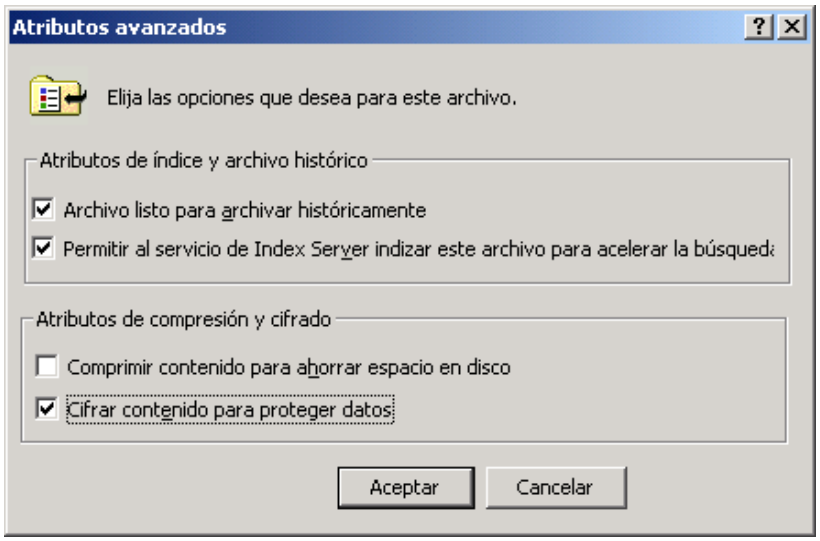
El problema con los sistemas de archivos sin cifrar es que si puede arrancar un volumen distante en la red o iniciar el servidor con otro sistema operativo (como con un disco de DOS o Windows) podrá leer el contenido del disco. Las restricciones de acceso contenidas en las listas de control de acceso no se aplicarán a los archivos contenidos en el disco.

Para cifrar un archivo o carpeta se debe seguir los siguientes pasos:

1. Abra el explorador de Windows. Haga clic con el botón derecho del mouse en el archivo o carpeta que desee cifrar y seleccione *propiedades*.
2. En la ficha general del cuadro de propiedades, haga clic en el botón *Avanzadas*.



3. En el cuadro de diálogo atributos avanzados, haga clic en la casilla *Cifrar contenido para proteger datos.*



4. Presione el botón *Aceptar.*

Ahora el archivo o carpeta se encuentra cifrado y almacenado en el disco.